



**CENTRO UNIVERSITÁRIO DE BRASÍLIA**  
**FACULDADE DE TECNOLOGIA E CIÊNCIAS SOCIAIS APLICADAS**  
**CURSO DE ENGENHARIA DA COMPUTAÇÃO**

**GUSTAVO SANTANA SANTOS**

**DESENVOLVIMENTO DE EQUIPAMENTO QUE MAXIMIZE A CAPTAÇÃO DE  
LUZ SOLAR**

**Orientadora: Prof<sup>a</sup>. M.C. Maria Marony Sousa Farias**

**BRASÍLIA**  
**1º SEMESTRE DE 2013**

**GUSTAVO SANTANA SANTOS**

**DESENVOLVIMENTO DE EQUIPAMENTO QUE MAXIMIZE A CAPTAÇÃO DE  
LUZ SOLAR**

**Trabalho apresentado ao UniCEUB – Centro  
Universitário de Brasília como pré-requisito  
para obtenção de Certificação de Conclusão do  
Curso de Engenharia de Computação.  
Orientadora: Prof<sup>a</sup>. M.C. Maria Marony Sousa  
Farias.**

**BRASÍLIA**  
**1º SEMESTRE DE 2013**

**GUSTAVO SANTANA SANTOS**

**DESENVOLVIMENTO DE EQUIPAMENTO QUE MAXIMIZE A CAPTAÇÃO DE  
LUZ SOLAR**

Trabalho apresentado ao UniCEUB – Centro  
Universitário de Brasília como pré-requisito  
para obtenção de Certificação de Conclusão do  
Curso de Engenharia de Computação.  
Orientadora: Prof<sup>a</sup>. M.C. Maria Marony Sousa  
Farias.

**Este Trabalho foi julgado adequado para a obtenção do Título de Engenheiro  
de Computação, e aprovado em sua forma final pela Faculdade de Tecnologia  
e Ciências Sociais Aplicadas - FATECS.**

---

**Prof. Abiezer Amarília Fernandes**  
**Coordenador do Curso**

**Banca Examinadora:**

---

**Prof<sup>a</sup>. Maria Marony Sousa Farias, mestre  
em Engenharia Elétrica – UFPB – PB.  
Orientadora**

---

**Prof. Flávio António Klein, mestre  
UniCEUB**

---

**Prof. William Oliveira Bessu, doutor  
em Engenharia de Estruturas  
UniCEUB**

---

**Prof. Luciano H. Duque, mestre  
UniCEUB**

*Dedico aos meus pais, que sempre foram exemplos e referências na minha vida. A minha esposa Roberta e aos meus filhos, Maria Eduarda e Gabriel.*

## **AGRADECIMENTOS**

Primeiramente a Deus. A toda minha família, que sempre buscaram o melhor para mim. Ao meu pai que gostaria imensamente que estivesse ao meu lado neste momento, pelo todo esforço por ele feito para que eu conseguisse chegar aqui.

E a minha esposa Roberta e meus filhos que com as dificuldades que encontrei durante o trabalho, me incentivaram a continuar e a concluir esta grande etapa da minha vida.

Muito Obrigado!

*“E se alguém ouvir as minhas palavras, e não crer, eu não o julgo; porque eu vim, não para julgar o mundo, mas para salvar o mundo.” **Jesus Cristo.***



## RESUMO

A população mundial vem se preocupando cada vez mais pela necessidade de aquisição de energia limpa, com a escassez do petróleo e seu alto impacto ao meio ambiente, torna a energia solar uma grande opção de outras formas de captação de energia, por ser de baixíssimo impacto ambiental. Esse trabalho tem como objetivo melhorar a captação de energia solar pelas placas fotovoltaicas que geralmente são construídas em bases fixas, o que diminuem consideravelmente sua capacidade de absorver a luz solar e aumentar sua eficiência energética. O Sistema proposto tem por objetivo fazer com que as placas fotovoltaicas possam acompanhar o movimento do curso do sol em seu deslocamento Leste-Oeste, potencializando consideravelmente a captação da luz solar. Foi necessário o estudo do movimento solar e os fenômenos de reflexão, junto com a construção de uma base móvel controlada por um micro controlador PIC, o qual comandava um servo motor tornando possível o acompanhamento da rota do sol pela placa aumentando sua eficiência de captação.

Palavras-chave: Energia solar, reflexão, Microcontrolador, Servo motor.

## **ABSTRACT**

The world population has been increasingly concerned by the need for the acquisition of clean energy, with the scarcity of oil and its high impact on the environment, makes solar energy a great option of other forms of energy, because it is very low environmental impact. This work aims to improve the capture of solar energy through solar panels, where these are usually fixed bases, which shortens their ability to absorb sunlight and increase their energy efficiency. With the goal of transforming these fixed bases in bases that can track the movement of the Sun, increasing considerably the capture of sunlight. It was necessary to study the solar movement and the phenomena of reflection, together with the construction of a mobile base controlled by a microcontroller PIC that do, which commanded a Servomotor making it possible to follow the route of the Sun through increasing its efficiency.

Keywords: solar Energy, reflection, microcontroller, Servomotor.

# SUMÁRIO

<b>LISTA DE FIGURAS.....</b>	<b>XIII</b>
<b>LISTA DE TABELAS E QUADROS.....</b>	<b>XV</b>
<b>LISTA DE ABREVIATURAS E SIGLAS.....</b>	<b>XVI</b>
<b>CAPÍTULO 1 – INTRODUÇÃO .....</b>	<b>17</b>
1.1 Introdução ao Tema Proposto .....	17
1.2 Motivação .....	18
1.3 Objetivos .....	19
1.4 Estrutura da Monografia.....	19
<b>CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA.....</b>	<b>21</b>
2.1 Acesso a energia elétrica no Brasil.....	21
2.2 Soluções Existentes .....	22
2.3 Benefícios do Dispositivo Proposto e Suas Restrições.....	22
<b>CAPÍTULO 3 – REFERENCIAL TEÓRICO.....</b>	<b>24</b>
3.1 Teoria de Rastreadores Solar .....	24
3.2 Radiação Solar a Nível do Solo .....	24
3.3 Movimento da Terra .....	25
3.4 Geometria Solar .....	27
3.5 Energia Solar Fotovoltaica.....	30
3.6 Efeito fotovoltaico.....	31
3.7 Tipos de Células.....	31
3.8 Microcontroladores .....	32
3.8.1 Microcontroladores da família PIC .....	33
3.9 Servomotor .....	34
3.10 Visão Geral do Projeto .....	35
<b>Capítulo 4 – DESCRIÇÃO DO HARDWARE E SOFTWARE.....</b>	<b>37</b>
4.1 Microcontrolador PIC 18F2450 .....	37
4.1.1 Especificações .....	38
4.1.2 Pinagem do PIC 18F2450.....	38
4.2 Placa Programadora ICD2 .....	41

4.3	MPLAB IDE.....	42
4.4	Servomotor .....	43
4.5	Placa Fotovoltaica .....	44
<b>CAPÍTULO 5 – IMPLEMENTAÇÃO.....</b>		<b>45</b>
5.1	Simulação do Movimento Solar .....	46
5.2	Elaboração do Circuito .....	47
5.3	Montagem do Circuito .....	48
5.4	Programação do Microcontrolador .....	49
5.5	Fluxograma Geral do Sistema .....	52
5.6	Estrutura do Protótipo .....	53
5.7	Montagem do Protótipo .....	57
<b>CAPÍTULO 6 – RESULTADOS OBTIDOS .....</b>		<b>58</b>
6.1	Simulações .....	58
6.1.1	Simulação Seguidor de Intensidade de Luz.....	58
6.1.2	Simulação Comparativa .....	58
6.2	Problemas Encontrados .....	59
6.3	Orçamento do Projeto .....	60
6.4	Protótipo Final.....	61
<b>CAPÍTULO 7 – CONSIDERAÇÕES FINAIS .....</b>		<b>62</b>
7.1	Conclusões .....	62
7.2	Propostas para Trabalhos Futuros .....	63
<b>REFERÊNCIAS BIBLIOGRÁFICAS .....</b>		<b>64</b>
<b>APÊNDICE A – CÓDIGO FONTE DO DISPOSITIVO.....</b>		<b>66</b>
<b>APÊNDICE B – BIBLIOTECA PWM.....</b>		<b>69</b>

## LISTA DE FIGURAS

Figura 1.1- Casa Solar Eficiente.....	09
Figura 3.1 - Componente da radiação solar ao nível do solo.....	15
Figura 3.2- Trajetória do Sol.....	16
Figura 3.3- Movimento de Translação da Terra.....	16
Figura 3.4 - Declinação Solar.....	18
Figura 3.5 - Ângulo de Azimute Solar.....	20
Figura 3.6- Efeito Fotovoltaico.....	21
Figura 3.7- Placa Fotovoltaica de Silício.....	22
Figura 3.8– Dimensionamento das famílias PIC e dsPIC.....	23
Figura 3.9- Servomotor.....	24
Figura 3.10- Tratamento de Sinais - Servomotor.....	25
Figura 3.11 – Diagrama esquemático do projeto.....	25
Figura 4.1– PIC 18F2450 utilizado no projeto.....	27
Figura 4.2 - Pinagem do PIC 18F2450.....	29
Figura 4.3– Pinagem do PIC 18F2450 utilizada no projeto.....	31
Figura 4.4 – Placa Programadora ICD2.....	32
Figura 4.5 – Programa MPLab IDE® .....	32
Figura 4.6 – Servomotor Motor Sat.....	33
Figura 4.7 - Especificações Placa Fotovoltaica SHSY54.....	34
Figura 5.1 – Protótipo Concluído.....	35
Figura 5.2 –Software de análise da movimentação Solar.....	36
Figura 5.3 – Modelo do Circuito.....	38
Figura 5.4 – Circuito do Projeto.....	39
Figura 5.5 – Plataforma MPLAB.....	39
Figura 5.6 – Fluxograma do Programa.....	42
Figura 5.7 – Primeiro modelo do Protótipo.....	43
Figura 5.8 – Projeção do Protótipo no AutoCAD.....	45
Figura 5.9 – Projeção do Protótipo desmontado no AutoCAD.....	45
Figura 5.10 – Software Impressora 3D.....	46

<b>Figura 5.11 – Impressão em Impressora 3D.....</b>	<b>46</b>
<b>Figura 5.12 – Montagem da Estruura.....</b>	<b>47</b>
<b>Figura 5.13 – Suporte das Placas Fotovoltaicas.....</b>	<b>47</b>
<b>Figura 5.14 - Protótipo finalizado.....</b>	<b>48</b>
<b>Figura 6.1 – Projeto Completo.....</b>	<b>52</b>

## LISTA DE TABELAS E QUADROS

Tabela 4.1 – Significado das nomenclaturas dos pinos do PIC18F2450.....	30
Tabela 4.2 - Especificações do servomotor.....	34
Tabela 4.3 - Especificações Placa Fotovoltaica Tabela 5.1 – Posição Norte-Sul.....	38
Quadro 6.1 – Comparativo de Tensão produzida.....	50
Quadro 6.2 – Custo do Projeto.....	51

## LISTA DE ABREVIATURAS E SIGLAS

<b><math>\Omega</math></b>	Ohms
<b>A/D</b>	Analógico / Digital
<b>ANEEL</b>	Agência Nacional de Energia Elétrica
<b>CEPEL</b>	Centro de Pesquisas de Energia Elétrica
<b>CRESESB</b>	Centro de Referência para Energia Solar e Eólica Sérgio de Salvo
<b><math>\mu\text{F}</math></b>	Micro-Farad
<b>LED</b>	Diodo Emissor de Luz
<b>PIC</b>	Programmable Interface Controller
<b>PWM</b>	Pulse Width Modulation
<b>SEESP</b>	Sindicato dos Engenheiros no Estado de São Paulo
<b>V</b>	Volts



## **CAPÍTULO 1 – INTRODUÇÃO**

### **1.1 Introdução ao Tema Proposto**

As fontes alternativas de energia têm sido cada vez mais exploradas pelas sociedades, grandes fenômenos climáticos demonstraram uma fraqueza de se depender exclusivamente de um meio de geração de energia elétrica. Falta de chuvas e mudanças constantes do clima fizeram, por exemplo, que o Brasil utilize-se de energia provida das usinas termoeletricas, o que aumentam consideravelmente os custos da geração de energia. Com o crescimento da economia brasileira, tem aumentado o consumo da população, considerando que automaticamente a população passa a consumir mais produtos eletrônicos e que a capacidade de geração de energia pelas hidroelétricas não crescem como o consumo da população, estamos caminhando para um possível colapso de fornecimento de energia elétrica no Brasil. (SEESP,2010).

A energia solar tem sido uma meio de fonte de energia bastante procurada por outros países, pois não necessita de grandes áreas para sua operação, e é insignificante no impacto ao meio ambiente. Como uma fonte não prejudicial ao meio ambiente, devido ser renovável, inesgotável e limpa, pode-se agregar a população um meio de fornecimento de energia em muitos casos se tornar a única fonte. Sua captação se dá por meios de painéis solares que podem ser utilizados para aquecimento de água, ou fotovoltaicos que se destinam a transformação da luz solar em energia elétrica. A eficiência de uma placa solar esta ligada diretamente a fatores como posicionamento, ângulo de incidência da luz solar e qualidade da placa coletora. (CRESESB, 2006).

Com o avanço da tecnologia, placas fotovoltaicas já são comercializadas com o custo bastante reduzido, porém sua utilização não explora 100% da capacidade de geração de energia, pois em grande maioria sua instalação é feita de forma fixa, tornando a pouco eficiente em grande parte do dia, pois a incidência de luz se diminui conforme o sol se movimenta perante a placa. Na figura 1.1 e mostrada uma instalação bastante comum em residências que utilizam placas fotovoltaicas, sua

posição e de forma fixa e impedindo uma eficiência constante da placa captadora. (CRESESB, 2006).



**Figura 1.1- Casa Solar Eficiente**

(FONTE: CEPEL, 2011)

Estudos realizados pela instituição CRESESB/CEPEL, que o ganho de uma estrutura de captação solar que fosse capaz de acompanhar o movimento do sol seria da ordem de 20%, um ganho bastante significativo quando se fala de energia. (CRESESB). Já estudos internacionais, colocam um aumento de até 40% ou mais na capacidade de geração de energia. (Renewable Energy World, 2009).

## **1.2 Motivação**

Este projeto teve como principal motivação as fortes discursões recentes sobre o tema de energia limpa e sustentabilidade. O tema Maximizador de captação da luz solar foi escolhido, por se tratar de um sistema o qual seu principal objetivo é aumentar o rendimento da produção de energia elétrica a partir do Sol, incentivando o uso da energia solar, que é fonte limpa e renovável. Outro fator que contribuiu fortemente na motivação deste projeto, foi a grande possibilidade de elaborar um protótipo que poderá agregar melhoria na vida de pessoas que ainda tem dificuldades no acesso a energia elétrica.

O Sol envia para a Terra energia equivalente a cerca de 10.000 vezes o consumo mundial de energia bruta. (CEPEL, 2011)

### **1.3 Objetivos**

Como objetivo geral este projeto tem a finalidade de apresentar um dispositivo de acompanhamento do curso do Sol, tornando uma placa fotovoltaica com inclinação próxima de 90 graus com a incidência de luz solar. Tornando o dispositivo uma meio independente que torna se viável sua instalação em qualquer meio, utilizando equipamentos computacionais simples e de baixo custo.

Os objetivos específicos deste trabalho, se resume aos seguintes tópicos:

- Coletar a incidência de luz solar em uma placa fotovoltaica e analisar o valor produzido.
- Desenvolver um circuito microcontrolado que processe as informações recebidas da placa, e com base nos dados coletados, ative um sistema que movimente uma painel fotovoltaico de acordo com o moviemto solar.
- Posicionar a placa no ponto de maior incidência de luz baseado na maior potência gerada pela placa fotovoltaica.

### **1.4 Estrutura da Monografia**

Esta monografia é dividida em seis capítulos, incluindo a INTRODUÇÃO, que trata da introdução ao tema proposto, a motivação do projeto e os principais objetivos. E essa seção que descreve toda a estrutura da monografia.

No segundo capítulo, APRESENTAÇÃO DO PROBLEMA, é apresentada uma descrição aprofundada sobre as questões motivacionais do trabalho, as soluções existentes atualmente sobre o tema proposto. E a última seção deste

capítulo aborda de forma resumida os benefícios do dispositivo proposto e suas restrições.

No terceiro capítulo, REFERENCIAL TEÓRICO, trata de assuntos como Servo motor, microcontroladores, Placas Fotovoltaicas, movimentação do Sol e efeitos de reflexão de forma conceitual. Nesse capítulo é apresentada também uma visão geral do projeto.

No quarto capítulo, DESCRIÇÃO DO HARDWARE E SOFTWARE, aborda as especificações dos dispositivos utilizados, é detalhada a especificação dos componentes microcontrolador PIC, Servo motor, Placa fotovoltaica e outros dispositivos e softwares.

No quinto capítulo, IMPLEMENTAÇÃO, é apresentado o desenvolvimento do projeto, como a integração entre o hardware e software, desenvolvimento do código na linguagem PIC e outras informações.

No sexto capítulo, RESULTADOS OBTIDOS abordam toda a parte de teste do protótipo, ajustes de problemas encontrados e suas dificuldades.

No sétimo capítulo, CONSIDERAÇÕES FINAIS, são apresentadas a conclusão e as sugestões para trabalhos futuros.

## **CAPÍTULO 2 – APRESENTAÇÃO DO PROBLEMA**

Este capítulo tem o objetivo de detalhar as questões que motivaram o trabalho, como: a criação de um dispositivo que maximize a captação da energia solar, que foi apresentado resumidamente na seção — Motivação — do capítulo anterior.

São apresentadas também algumas soluções existentes para estas questões, no que diz respeito à Captação de energia solar. E por fim, são apresentados os benefícios do dispositivo proposto pelo trabalho e suas restrições.

### **2.1 Acesso a energia elétrica no Brasil**

Levantamento inédito feito pela Agência Nacional de Energia Elétrica (Aneel) junto às distribuidoras de energia apontou que o Brasil ainda possui um milhão de residências sem luz. O número é muito maior do que o estimado pelo governo federal, que indicava em setembro, com base em dados do Censo de 2010, apenas 378 mil residências ainda sem energia elétrica. Com os novos números apresentados, as distribuidoras pediram prazos mais elásticos para universalizar o acesso no país, meta que a presidente Dilma Rousseff fixou para o fim de 2014. Algumas, como as de Mato Grosso e Tocantins, querem jogar a meta para 2027. (O Globo, 2012)

O parágrafo acima nos mostra a realidade que um grande número de Brasileiros ainda não tem acesso a energia elétrica no Brasil. Segundo a ANEEL 75,9% da energia elétrica é proveniente de usinas hidroelétricas, o que torna difícil a distribuição de energia elétrica em lugares mais remotos e deslocados de grandes meios. Parte dessa população vivem em áreas rurais, com difícil acesso a redes elétricas, com um custo muito elevado para o provimento de energia elétrica nessas áreas.

A População indígena também sofre com o acesso a energia elétrica no Brasil, pois em sua maioria eles estão instalados em meio a florestas e vales com grandes dificuldades de locomoção e instalações de torres.

## 2.2 Soluções Existentes

Existem vários modelos de seguidores solares, os quais podem ser classificados de diversas formas. O Instituto de Eletrotécnica e Energia da USP classificou os seguidores solares de acordo com o número de eixos, tipo de estrutura, tipo de controle e estratégia de rastreamento (iee.usp.br, 2012).

- Seguidores de Único Eixo – Nesse modelo as placas realizam apenas um movimento para acompanhamento do curso do sol, geralmente movimento Leste-Oeste e seu mecanismo de busca é mais simples e um custo menor para serem construídos.
- Seguidores de Duplo Eixo – Nesse modelo as placas realizam dois movimentos para acompanhamento do curso do sol, o movimento Leste-Oeste e movimento Norte-Sul. Seu mecanismo de busca é mais complexo e um custo maior para serem construídos.
- Seguidores do Tipo Parabólica – Nesse modelo a placa fotovoltaica ficam na posição oposta ao sol e uma coxa fica no sentido oposto ao da placa e de frente para o sol, dessa forma os raios solares são refletidos pela coxa direcionados para o centro da placa. Seu mecanismo de busca é mais complexo, custo maior para serem construídos e são utilizados para grandes estruturas.

## 2.3 Benefícios do Dispositivo Proposto e Suas Restrições

O objetivo principal do projeto é a apresentação de um dispositivo de que realiza a movimentação de uma placa fotovoltaica para que essa possa ficar sempre voltada perpendicularmente aos raios solares, dessa forma terá como benefício uma maior produção de energia, pois a placa trabalharam sempre em sua plenitude na produção de energia elétrica, e não será impactada com o deslocamento do sol como ocorre com as placas instaladas de forma fixa.

O projeto não tem como funcionalidade o tratamento e armazenamento da energia gerada. O protótipo somente fará o acompanhamento do curso do Sol, conforme estudo do comportamento a incidência solar na superfície terrestre e o ângulo de incidência.

## **CAPÍTULO 3 – REFERENCIAL TEÓRICO**

Neste capítulo é exibida uma visão de forma geral do projeto proposto. Por isso é essencial para o entendimento do mesmo a compreensão dos assuntos abordados neste capítulo.

Para o melhor entendimento deste trabalho, é necessário o estudo de conceitos teóricos relacionados ao projeto desenvolvido. Embora alguns dos conceitos citados exijam um detalhamento mais amplo do tema para sua total compreensão, visando o foco principal, serão apresentadas apenas as características mais relacionadas ao projeto.

### **3.1 Teoria de Rastreadores Solar**

Estudos acadêmicos e de fornecedores de placas fotovoltaicas, mostram a melhoria na produtividade das placas em uma faixa de valor aproximado de 25% a 40% com o uso de rastreadores solares. Esses rastreadores tem como função direcionar o painel para uma posição perpendicular a incidência da luz do Sol. (<http://solnaamazonia.com.br>, 2011)

### **3.2 Radiação Solar a Nível do Solo**

De toda a radiação solar que chega às camadas superiores da atmosfera, apenas uma fração atinge a superfície terrestre, devido à reflexão e absorção dos raios solares pela atmosfera. Esta fração que atinge o solo é constituída por um componente direta e por uma componente difusa. Conforme ilustrado na figura 3.1.



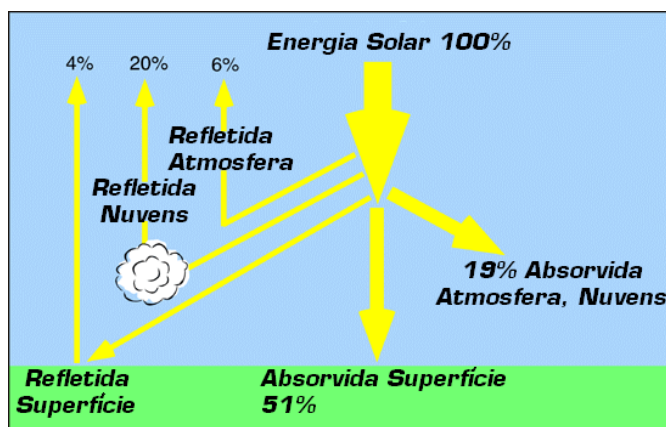


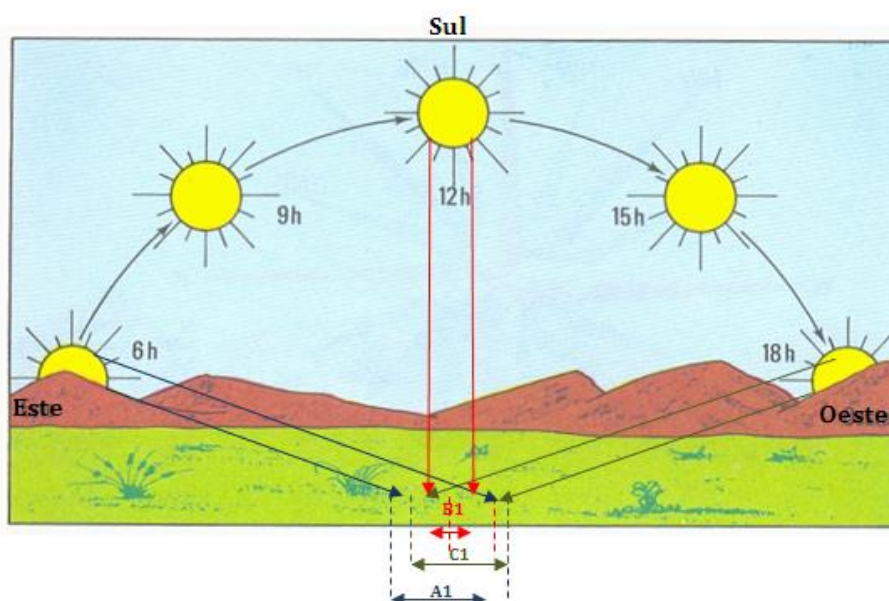
Figura 3.1 - Componente da radiação solar ao nível do solo  
(FONTE:CRESESB, 2006)

Na Figura 3.1 é ilustrada uma situação que ocorre na Terra. Nela podem ser observados os fenômenos de reflexão. Os percentuais apresentados na figura variam de acordo com a incidência solar. Diversos fatores podem mudar estes percentuais; é o caso das estações do ano que está relacionado ao movimento de translação que será discutido a seguir. [MORAES]

### 3.3 Movimento da Terra

Os principais movimentos da Terra – rotação em torno do próprio eixo e translação ao redor do Sol – são percebidos como movimentos aparentes do Sol. A inclinação do eixo da Terra (Aprox.  $23.27^\circ$ ) em relação ao plano de translação causam as estações do ano e a variação na duração dos dias (período de insolação).

Graças ao movimento de rotação, o Sol aparece (nascente) de um lado (leste) pela manhã e desaparece (poente) à tarde no outro lado (oeste). Parece que o Sol está se movimentando, quando na verdade é a Terra que se move. Esse fenômeno é chamado de MOVIMENTO APARENTE DO SOL. Conforme apresentado na figura 3.2, pode-se observar que ao longo do dia, o sol muda de posição com o passar das horas. (JOSELI, 2011)



**Figura 3.2- Trajetória do Sol**

(FONTE: <http://joselimaia.blogspot.com.br>, 2011)

Na Figura 3.3 é ilustrado o movimento de translação, onde as estações do ano podem ser observadas; estando essas definidas por uma data de início e fim. Onde o movimento elíptico e a inclinação do eixo da terra (Aprox.  $23.27^\circ$ ), tem sua influência. Cada estação abrange um período de 3 meses.



**Figura 3.3- Movimento de Translação da Terra**

(FONTE: <http://www.alunosonline.com.br>, 2011)

### 3.4 Geometria Solar

Com base dos estudos dos movimentos solares e seu comportamento cíclico, foi possível determinar a radiação solar que atinge uma superfície. Conforme definido por Duffie, John A., o fluxo de radiação solar é determinado por cálculo de três componentes básicas, a radiação direta do Sol, a radiação difusa e radiação refletida pelo entorno. A fórmula para se determinar a radiação em uma determinada superfície, é expressa pela fórmula (1):

$$I_T = I_{DN} \cdot \cos \theta + I_d + I_r$$

(1)

Onde:

( $I_T$ ) = fluxo total de radiação que atinge a superfície (W/m<sup>2</sup>);

( $I_{DN}$ ) = fluxo de radiação direta do sol (W/m<sup>2</sup>);

( $\theta$ ) = ângulo de incidência dos raios solares sobre uma superfície (°);

( $I_d$ ) = fluxo de radiação difusa do céu (W/m<sup>2</sup>);

( $I_r$ ) = fluxo de radiação refletida (W/m<sup>2</sup>).

O valor da radiação solar direta é obtido conforme expressão da fórmula (2):

$$I_{DN} = A \cdot e^{-(B/\sin \beta)}$$

(2)

onde:

( $I_{DN}$ ) = fluxo de radiação direta do sol (W/m<sup>2</sup>);

$\beta$ : altitude solar (ângulo entre o plano horizontal sobre a terra e os raios solares) (°);

A: radiação solar aparente para massa de ar igual a zero (tabelada) (W/m<sup>2</sup>);

B: coeficiente de extinção atmosférica (tabelada) adimensional;

$\varphi$ : azimuth solar (ângulo entre a direção Sul e a projeção dos raios solares no plano horizontal sobre a terra) (°).

A declinação solar é dado pelo ângulo entre a linha do Equador e o eixo que sai do centro do Sol, como pode ser observado na Figura 3.4.



**Figura 3.4 - Declinação Solar**

(Fonte: <http://www.newhome.com.br>)

Devido à inclinação do eixo da terra ser de aproximadamente de 23,27° sobre o plano da órbita à volta do Sol, a declinação terrestre pode variar entre 23,27° ao norte e 23,27° a sul durante o movimento de translação. Esse valor varia conforme o dia do ano, e pode ser calculado a partir da resolução da fórmula (3), onde N será o dia do ano, contado a partir do dia 1º de Janeiro. (<http://pcc261.pcc.usp.br>, 2006).

$$\delta = 23,47 \cdot \sin \left( 360 \cdot \frac{284 + N}{365} \right)$$

(3)

onde:

( $\delta$ ) = ângulo de declinação solar (°);

N = é o dia do ano contado a partir de 1º de janeiro.

Para determinar o ângulo de incidência dos raios solares sobre uma superfície em uma latitude, é necessário obter primeiramente o ângulo de Azimute. A fórmula (4) é utilizada para a determinação deste valor. Como o ângulo de Azimute é determinado pelo plano horizontal da terra, o valor do Azimute utilizado

para o cálculo do ângulo de inclinação deverá ser a diferença entre o ângulo de Azimute da superfície.

$$\cos \theta = \cos \beta \cdot \cos \gamma \cdot \sin \alpha + \sin \beta \cdot \cos \alpha$$

(4)

onde:

$\theta$  = ângulo de incidência dos raios solares sobre uma superfície (°);

$\beta$  = altitude solar já definida;

$\alpha$  = inclinação da superfície em relação ao plano horizontal (°);

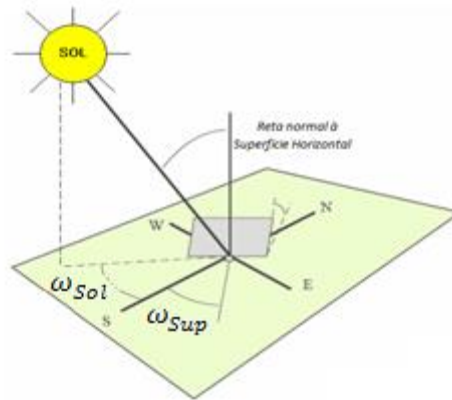
$\gamma$  = diferença entre o azimute solar e o da superfície (°);

A diferença entre o ângulo de azimute solar e o ângulo da superfície é dado por:

$$\gamma = \omega_{Sup} \mp \omega_{Sol}$$

(5)

O ângulo de Azimute solar é o ângulo determinado pela posição de um astro em relação ao eixo norte-sul, sendo positivo em direção ao oeste e negativo em direção ao leste, como demonstrado na Figura 3.5. A fórmula (6), é a fórmula utilizada para se obter o valor do ângulo de Azimute do sol (Duffie, John A., Beckman, Willian A., 1991).



**Figura 3.5 - Ângulo de Azimute Solar**

(Fonte: [http:// http://www.electronica-pt.com/](http://www.electronica-pt.com/))

$$\cos \theta = \frac{\sin \beta \cdot \sin L - \sin \delta}{\cos \beta \cdot \cos L}$$

(6)

onde:

$\theta$  = ângulo de Azimute solar (°);

$\beta$  = altitude solar já definida;

L: latitude (negativo no hemisfério sul) (°);

$\delta$  = ângulo de declinação solar(°);

### 3.5 Energia Solar Fotovoltaica

A Energia Solar Fotovoltaica é a energia gerada através da conversão da luz em energia elétrica, denominado efeito fotovoltaico (NASCIMENTO, 2004). O efeito fotovoltaico, relatado por Edmond Becquerel, em 1839, trata-se do aparecimento de uma diferença de potencial nos extremos de uma estrutura de um material semicondutor, através da absorção da luz. Para que esse fenômeno ocorra, a célula fotovoltaica é fundamental no processo de conversão. (CRESESB, 2006).

### 3.6 Efeito fotovoltaico

O efeito fotovoltaico é a junção de materiais semicondutores do tipo PN, onde o objetivo é a criação de um campo elétrico no interior do material, de forma que as cargas negativas e as positivas fiquem separadas cada uma em um lado. O cristal de silício é o mais utilizado para construção de placas fotovoltaicas, porém o cristal puro é um mal condutor, pois não possui em sua composição elétrons livres. Para que o silício se torne um material capaz de reagir aos efeitos da luz, é realizado um processo conhecido como dopagem. A junção de uma célula solar de silício é constituída por uma parte dopada com fósforo (P), denominada de tipo n, e outra parte dopada com boro (B), denominada de tipo p. é na fronteira destas duas partes que se cria um campo elétrico para separar as cargas positivas das negativas. (CRESESB, 2006). A Figura 3.6 abaixo, simula o efeito fotovoltaico através do funcionamento de uma placa.

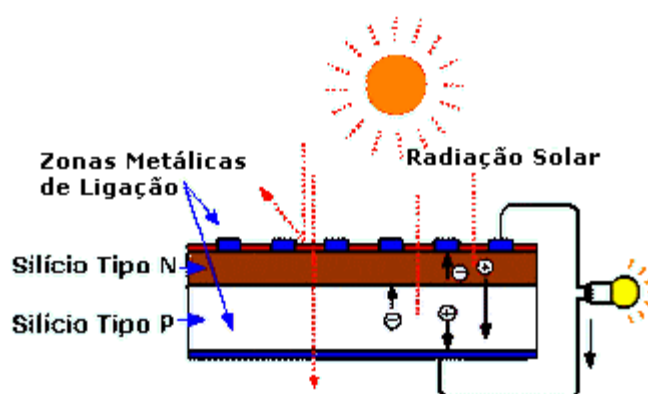
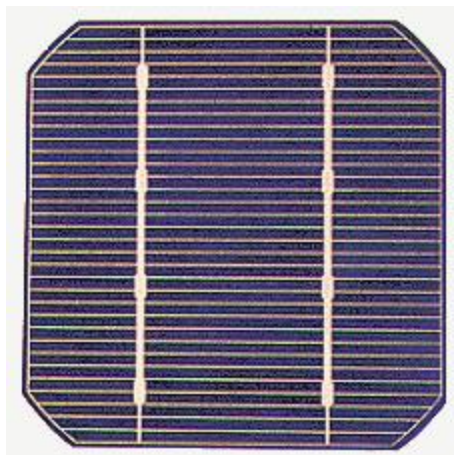


Figura 3.6- Efeito Fotovoltaico

(FONTE: <http://www.electronica-pt.com>, 2011)

### 3.7 Tipos de Células

As células fotovoltaicas são fabricadas, na sua grande maioria, usando o silício (Si) e podendo ser constituída de cristais monocristalinos, policristalinos ou de silício amorfo. A figura 3.7 representa uma célula fotovoltaica de Silício Cristalino, um modelo de fácil disponibilidade no mercado. (CRESESB, 2006)



**Figura 3.7- Placa Fotovoltaica de Silício**

(FONTE: CRESESB, 2006)

### **3.8 Microcontroladores**

O microcontrolador é um dispositivo semicondutor em forma de circuito integrado, que integra as partes básicas de um microcomputador - microprocessador, memórias não-voláteis e voláteis e portas de entrada e saída. Geralmente, é limitado em termos de quantidade de memória, principalmente no que diz respeito à memória de dados, é utilizada em aplicações específicas, ou seja, naquelas que não necessitam armazenar grandes quantidades de dados. Apresenta um custo bastante baixo e possui vários fornecedores. (GIMENEZ, 2002, p.4).

Alguns dos fornecedores mencionados acima são: Hitachi, Intel, Microchip, Mitsubishi, Motorola, Philips, Toshiba etc.

Foi utilizado neste projeto o microcontrolador PIC18F2450, por ter uma grande quantidade de pinos de entrada e saída (23 pinos dos 28 pinos do microcontrolador), por ter um fácil acesso no comércio local e principalmente porque os microcontroladores da série 18F têm conversor analógico/digital, modulo PWM (Pulse Width Modulation), Biblioteca com varias funções para controle de periféricos disponível e com fácil programação na linguagem C.



### 3.8.1 Microcontroladores da família PIC

O microcontrolador PIC produzido pela *Microchip Technology Inc.*, é um circuito integrado que em um único dispositivo possui todos os circuitos necessários para realizar um sistema programável completo.

Os microcontroladores da família PIC possuem modelos variados com características análogas e que podem ser operados por um parecido set de instruções. Embora apresentando características principais parecidas, eles se distinguem quanto a seus periféricos (memórias, comparadores, conversores) e quantidade de pinos. Dessa forma, há uma diversidade de microcontroladores e funcionalidades. (SOUZA, 2005)

A denominação família PIC é devido aos vários microcontroladores fornecidos, que variam de 8-bits, 16-bits até 32-bits, conforme mostrada na Figura 3.1.

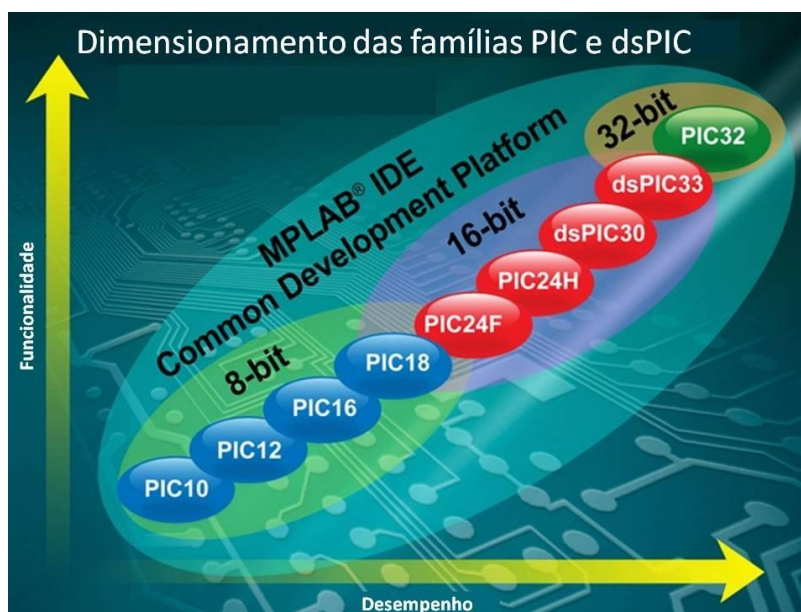


Figura 3.8– Dimensionamento das famílias PIC e dsPIC.  
(Fonte: MICROCHIP TECHNOLOGY INC, adaptada).

As principais características técnicas do microcontrolador PIC18F2450, serão abordadas na seção – Microcontrolador PIC18F2450- no capítulo a seguir.

### 3.9 Servomotor

Um servomotor é um motor com um circuito de controle de posição acoplado. Os servomotores não dão uma volta completa em seu eixo, eles trabalham em uma faixa ou de 90 ou 180 graus em seu eixo.

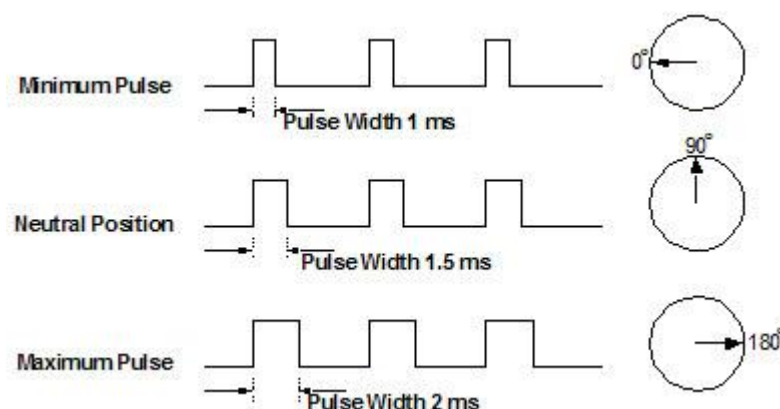
Internamente o servomotor consiste de um circuito de controle que recebe um sinal de controle para que ele se posicione em um determinado ângulo. A figura 3.9 abaixo, detalha um circuito e engrenagens de um servomotor.



**Figura 3.9- Servomotor**

(FONTE: <http://www.pictronics.com.br>)

O servo é ligado a uma saída de um microcontrolador, por exemplo, e recebe um sinal no formato PWM (Pulse Width Modulation). Este sinal pode ter 0Volts ou 5Volts. O circuito de controle do servo fica monitorando este sinal em intervalos de 20ms (milissegundos), se dentro deste intervalo ele percebe uma alteração do sinal de 0v para 5v durante 1ms até 2ms ele altera a posição do seu eixo para coincidir com o sinal que recebeu.(Pictronics, 2010). A figura 3.10 simula esse funcionamento do servomotor.



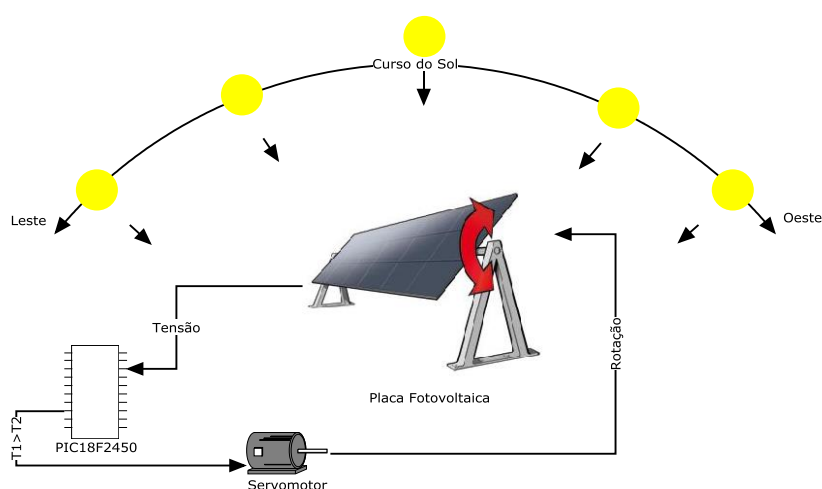
**Figura 3.10- Tratamento de Sinais - Servomotor**

(FONTE: <http://www.pictronics.com.br>)

### 3.10 Visão Geral do Projeto

O projeto basicamente é constituído por um microcontrolador (PIC do modelo 18F2450), Placa fotovoltaica, um servomotor e demais componentes eletrônicos, tais como resistores, capacitores, oscilador de cristal, led's, diodos e outros. O diagrama esquemático do projeto proposto pode ser observado na Figura 3.11, na qual representa de forma objetiva a composição geral do projeto.

O diagrama esquemático do projeto é ilustrado na Figura 3.11.



**Figura 3.11 – Diagrama esquemático do projeto (FONTE: Autor)**

- 1) Captação da Luz Solar pelas Placa Fotovoltaicas;
- 2) Através da entrada AD do Microcontrolador e realizado a leitura da tensão da energia produzida pelas placas;

- 3) O Microcontrolador irá comparar o valor da Tensão captado anteriormente e compara-lo com a atual.
- 4) Após a leitura da tensão atual, será comparado com a leitura anterior, caso seja maior que a anterior, o microcontrolador irá movimentar o Servomotor com um pulso;
- 5) Microcontrolador realizará o ciclo até que se encontre o ponto de máxima tensão produzido pelas placas fotovoltaicas;

Detalhes sobre o *hardware* e o *software* utilizados e seus funcionamentos encontram-se no CAPÍTULO 4 — DESCRIÇÃO DE HARDWARE E SOFTWARE. A implementação do projeto pode ser observada no CAPÍTULO 5 — IMPLEMENTAÇÃO.

## CAPÍTULO 4 – DESCRIÇÃO DO HARDWARE E SOFTWARE

Este capítulo detalha as especificações dos dispositivos usados e seu devido funcionamento nesse projeto, tanto a parte física, ou seja, o *hardware*, como a parte lógica, os *softwares* utilizados.

### 4.1 Microcontrolador PIC 18F2450

O microcontrolador utilizado nesse projeto foi o PIC 18F2450, conforme mencionado no capítulo anterior. Ele é o componente responsável em realizar toda a logica que foi programada, ele irá manter a comunicação com o circuito, onde estão ligados os outros componentes do projeto.

A Figura 4.1 ilustra o PIC 18F2450 utilizado neste projeto.



Figura 4.1– PIC 18F2450 utilizado no projeto (Fonte:Microchip)

### 4.1.1 Especificações

As principais especificações técnicas do PIC 18F2450 são: (Microchip).

- Microcontrolador de 28 pinos;
- Programação *in-circuit* (alta e baixa tensão);
- Frequência de operação de até 40 MHz;
- Memória *flash* de programa de 8kwords;
- Memória de programa: Flash de 16 Kb
- Memória de dados: SRAM de 768 bytes
- Oscilador interno de 31KHz
- Oscilador externo de 48 Mhz
- 23 Terminais de I/O que suportam correntes de até 25 mA
- 2 Temporizadores de 16 bits y 1 Temporizador de 8 bits
- Módulos de comunicação serie (EUSART), PWM, CCP
- Módulo de comunicação USB 2.0
- 10 canais A/D de 10 bits

Este modelo da família PIC foi escolhido devido ao fato de possuir 23 portas configuráveis de entrada e saída, comunicação serial, conversor A/D e módulo PWM e CCP, conforme estas características a elaboração deste dispositivo foi possível.

### 4.1.2 Pinagem do PIC 18F2450

A pinagem do PIC 18F2450 é mostrada na Figura 4.2 e o significado das nomenclaturas utilizadas na identificação desses pinos na Tabela 4.1.

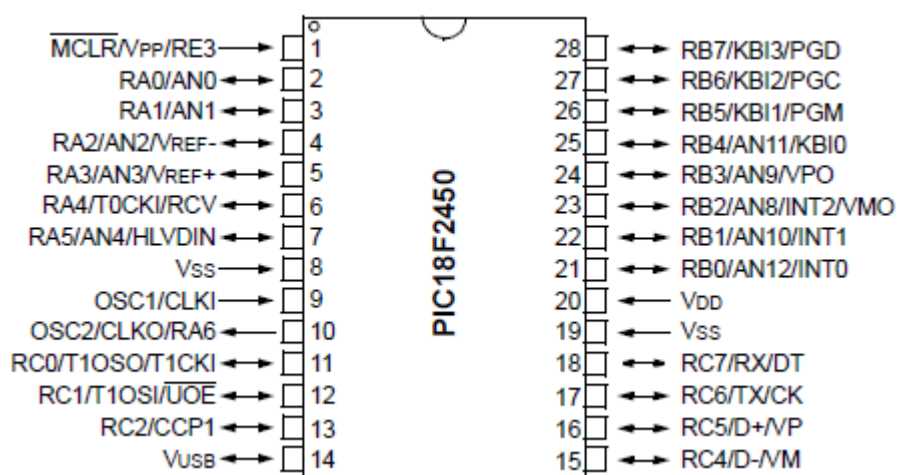


Figura 4.2 - Pinagem do PIC 18F2450

(Fonte: Microchip Technology Datasheet, 2003, p.3)

Tabela 4.1 – Significado das nomenclaturas dos pinos do PIC18F2450.

N°	Pino	Descrição
1	MCLR/Vpp/RE3	Master Clear (reset). O microcontrolador funciona quando este pino está em nível alto.
2	RA0/ AN0	Entrada e saída digital. Entrada analógica.
3	RA1/ AN1	Entrada e saída digital. Entrada analógica.
4	RA2/ AN2/ V <sub>REF</sub>	Entrada e saída digital. Entrada analógica. Tensão negativa de referência analógica.
5	RA3/ AN3/ V <sub>REF+</sub>	Entrada e saída digital. Entrada analógica. Tensão positiva de referência analógica.
6	RA4/ T0CKI/ RCV	Entrada e saída digital. Open-drain quando configurado como saída. Entrada externa do contador TMR0. Saída do comparador 1.
7	RA5/ AN4/ SS/ C2OUT	Entrada e saída digital. Entrada analógica. Slave para a comunicação SPI. Saída do comparador 2.
8	RE0/ RD/ AN5	Entrada e saída digital. Controle de leitura da comunicação paralela. Entrada analógica.

9	RE1	Entrada e saída digital
	WR	Controle de escrita da comunicação paralela.
	AN6	Entrada analógica.
10	RE2	Entrada e saída digital.
	CS	Habilitação externa para comunicação paralela.
	AN7	Entrada analógica.
11/32	VDD	Alimentação positiva.
12/31	VSS	GND.
13	OSC1/CLKIN	Oscilador cristal ou entrada de osciladores externos.
14	OSC2/CLKOUT	Saída para oscilador cristal.
15	RC0	Entrada e saída digital.
	T1OSO	Saída do oscilador externo para TMR1.
	T1CKI	Entrada de incremento para TMR1.
16	RC1	Entrada e saída digital.
	T1OSI	Entrada do oscilador externo para TMR1.
	CCP2	Entrada do Capture2 ou Saída para Compare2/PWM2.
17	RC2	Entrada e saída digital.
	CCP1	Entrada do Capture1 ou Saída para Compare1/PWM1.
18	RC3	Entrada e saída digital.
	SCK	Entrada/Saída do clock para comunicação SPI.
	SCL	Entrada/Saída do clock para comunicação I2C.
19	RD0	Entrada e saída digital.
	PSP0	Comunicação paralela.
20	RD1	Entrada e saída digital.
	PSP1	Comunicação paralela.
21	RD2	Entrada e saída digital.
	PSP2	Comunicação paralela.
22	RD3	Entrada e saída digital.
	PSP3	Comunicação paralela.
23	RC4	Entrada e saída digital.
	SDI	Entrada de dados para comunicação SPI.
	DAS	Entrada/Saída de dados para comunicação I2C.
24	RC5	Entrada e saída digital.
	SDO	Saída de dados para comunicação SPI.
25	RC6	Entrada e saída digital.
	TX	Transmissão para comunicação assíncrona USART.
	CK	Clock para comunicação síncrona USART.
26	RC7	Entrada e saída digital.



	RX DT	Recepção para comunicação assíncrona USART. Dados para comunicação síncrona USART.
27	RD4 PSP4	Entrada e saída digital. Comunicação paralela.
28	RD5 PSP5	Entrada e saída digital. Comunicação paralela.

A pinagem escolhida no projeto pode ser observada na Figura 4.3. Essa configuração atende às necessidades de cada componente utilizado no projeto

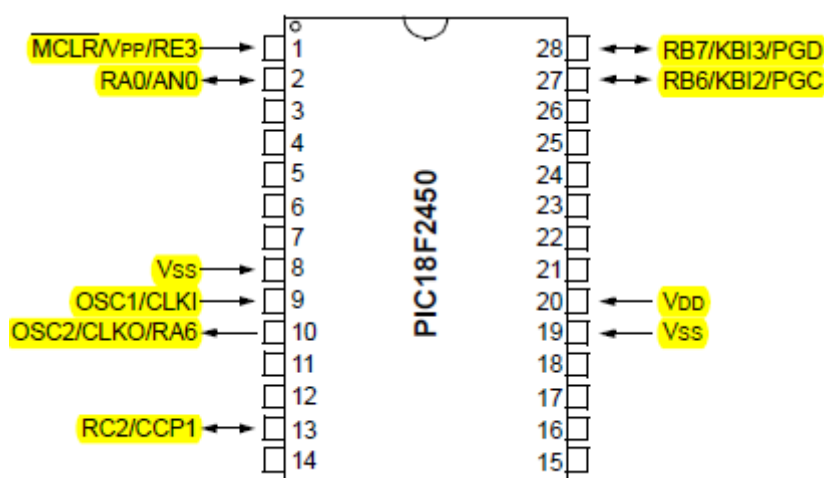


Figura 4.3– Pinagem do PIC 18F2450 utilizada no projeto.

(Fonte: Microchip, modificada)

## 4.2 Placa Programadora ICD2

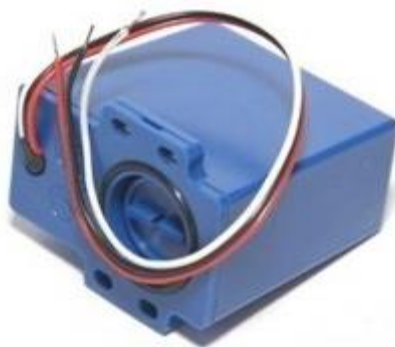
A placa programadora ICD2 é um dispositivo que permite comunicar o computador diretamente com o PIC. É composto do conector ICSP (In-Circuit Serial Programming) para gravação em microcontroladores PIC de 8, 14, 18, 20, 28 e 40 pinos. (MICROCHIP)

A grande vantagem de usar a placa programadora ICD2 com conector ICSP, é a possibilidade de se gravar o código no microcontrolador sem a retirada do mesmo do circuito. Isso agiliza os testes e a alteração em tempo real do código desenvolvido para ajustes no programa. O código é compilado e enviado do PC para a Placa Programadora via USB e o Software que faz essa transferência é o MPLAB IDE, detalhado no próximo tópico. (MICROCHIP)



## 4.4 Servomotor

Um servomotor de giro contínuo fabricado pela Chaparral Electronics, modelo Motor Sat, mostrado na Figura 4.6, foi utilizado para movimentar o painel fotovoltaico. Na tabela 4.2 estão as especificações técnicas do modelo utilizado. O Servomotor foi escolhido devido ser um equipamento de baixo custo, alta qualidade e alto torque.



**Figura 4.6 – Servomotor Motor Sat.**

(Fonte: <http://www.antenasmundosat.com.br/>)

**Tabela 4.2 - Especificações do servomotor**

SERVO Motor Sat		
Tensão de alimentação	5 V DC	
Corrente	500 mA	
Tempo de Período	17 – 21 mseg	
Temperatura	-20° C	+60° C
Tamanho do Pulso	0,8 – 2.2 msec.	
Tamanho	29,5 mm x 20,0 mm x 35,6 mm	
Peso	56 g	

## 4.5 Placa Fotovoltaica

A placa fotovoltaica utilizada no projeto, conforme mostrado na Figura 4.7, foi utilizada para realizar a captação da energia sol e identificar o maior ponto de tensão produzido. Na tabela 4.3 estão as especificações técnicas do modelo utilizado.



**Figura 4.7 - Especificações Placa Fotovoltaica SHSY54**

(Fonte: <http://www.dx.com>)

**Tabela 4.3 - Especificações Placa Fotovoltaica SHSY54**

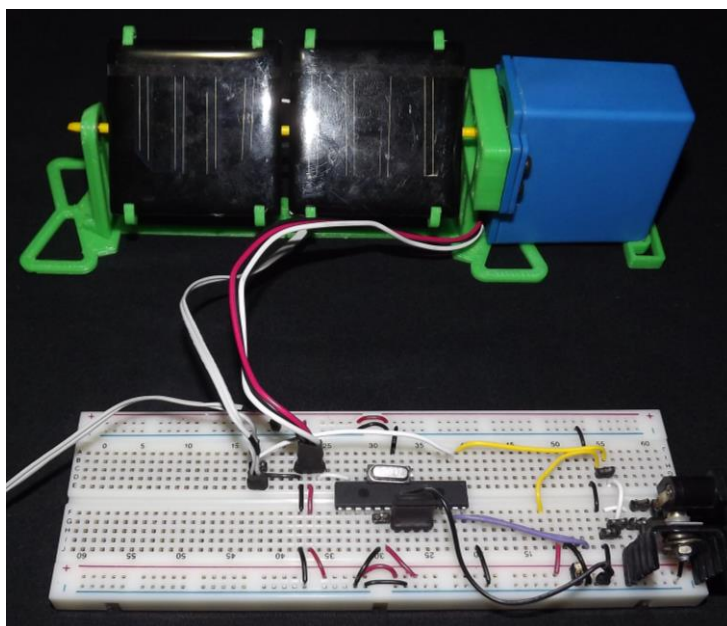
Placa Fotovoltaica SHSY54	
Tensão Máxima Gerada	4V
Máximo fornecimento de corrente	80mA
Dimensões	5.4cm x 5.4cm x 0.3cm
Peso	17g

## CAPÍTULO 5 – IMPLEMENTAÇÃO

Neste capítulo são apresentados tópicos fundamentais para compreensão geral da implementação do projeto, foram definidas as etapas necessárias para isso, que são:

- Simulação do movimento Solar;
- Elaboração do circuito;
- Montagem do circuito;
- Programação do Microcontrolador;
- Estrutura do Protótipo;
- Montagem do protótipo;

Na Figura 5.1 é mostrado o protótipo em sua fase final.



**Figura 5.1 – Protótipo Concluído (Fonte: Autor)**

## 5.1 Simulação do Movimento Solar

Para analisar e simular o movimento aparente do sol, fatores que interferem diretamente na radiação solar sobre a superfície o painel fotovoltaico, foi utilizado um software chamado **Motions of the Sun Simulator**. Com a utilização deste software, foi possível simular o movimento aparente do sol baseado em diferentes latitudes, facilitando a definição e planejamento do movimento que o painel fotovoltaico irá realizar. Conforme apresentado na figura 5.2, foi possível identificar quais as angulações no sentido Norte-Sul o protótipo deveria ter para um melhor aproveitamento da captação da luz. Dessa forma foi possível maximizar a captação juntamente com o movimento Leste-Oeste que o Painel realiza.

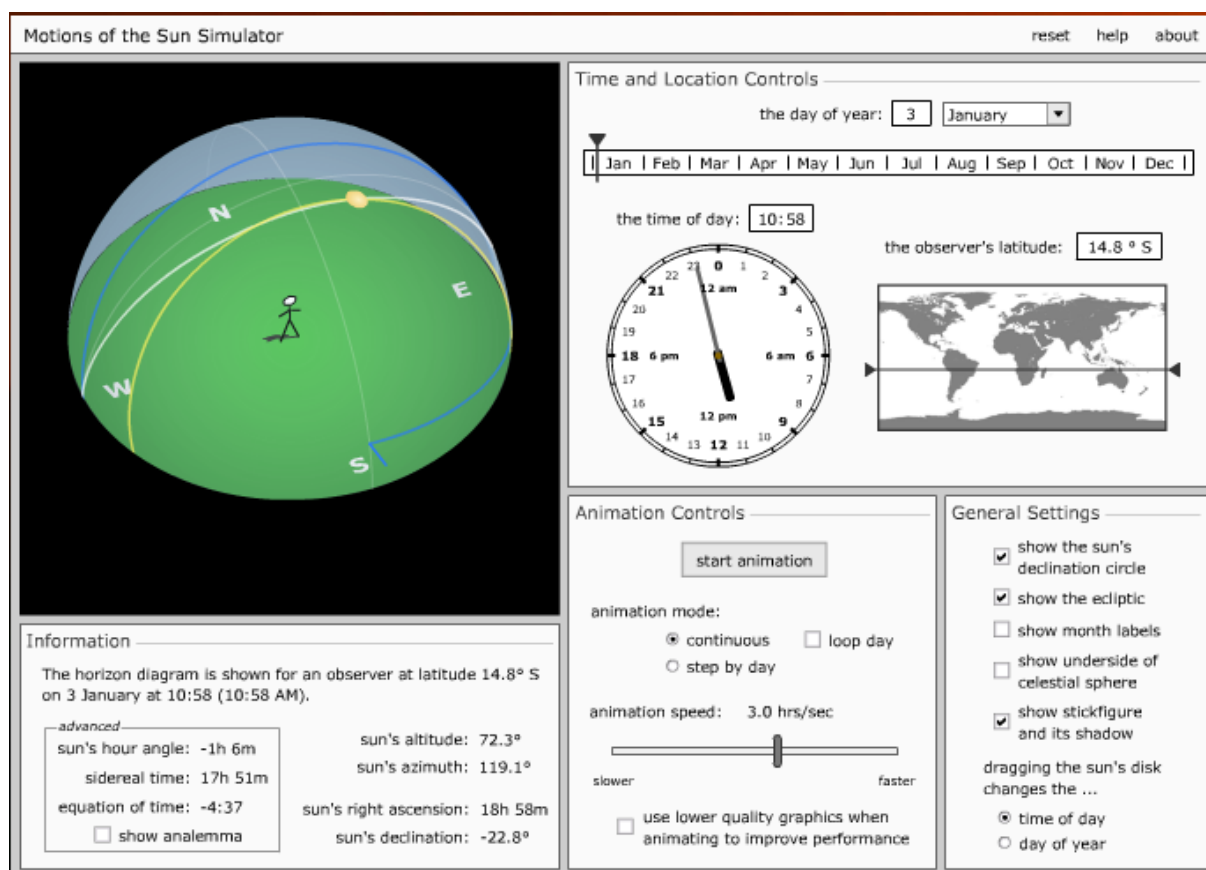


Figura 5.2 –Software de análise da movimentação Solar

(FONTE: <http://astro.unl.edu/>)

A tabela 5.1 apresenta a inclinação Norte-Sul que o protótipo deverá ser posicionado para uma melhor captação da incidência da luz solar, já que essa movimentação não é contemplada pelo protótipo. Essa angulação é gerada pelo Software Motions of the Sun Simulator.

**Tabela 5.1 – Posição Norte-Sul**

<b>Mês</b>	<b>Altura do Sol ( Sun Altitude )</b>	<b>Inclinação ( Array Tilt )</b>	<b>Inclinar para o:</b>
JAN	85	5	Sul
FEV	86	4	Norte
MAR	75	15	Norte
ABR	63	27	Norte
MAI	55	35	Norte
JUN	52	38	Norte
JUL	55	35	Norte
AGO	63	27	Norte
SET	75	15	Norte
OUT	87	3	Norte
NOV	85	5	Sul
DEZ	82	8	Sul

Fonte: [www.heliodon.com.br](http://www.heliodon.com.br)

## 5.2 Elaboração do Circuito

O Primeiro passo realizado no desenvolvimento do protótipo foi à criação do circuito através do software PCAD, onde foi possível planejar e dimensionar todos os componentes necessários para o desenvolvimento do sistema. Foram projetadas todas as conexões que o circuito realizaria com o microcontrolador PIC18F2450, como pode ser observado na Figura 5.3. Durante o desenvolvimento do projeto o modelo do circuito foi ajustado, para atender necessidades do modelo real.

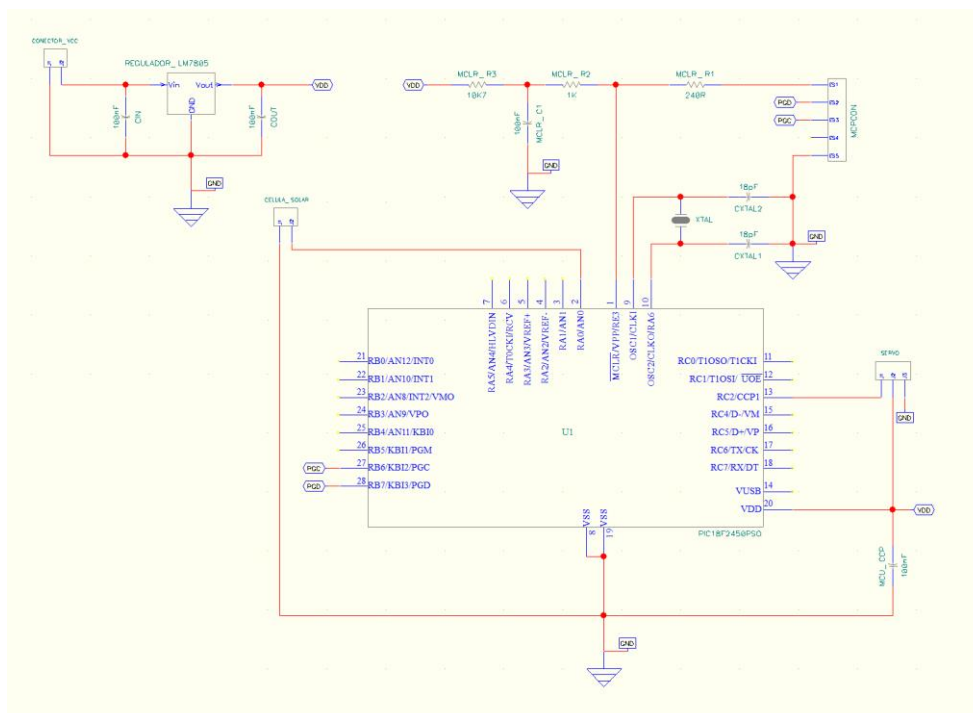


Figura 5.3 – Modelo do Circuito (Fonte: Autor)

### 5.3 Montagem do Circuito

Em uma Protoboard MP 830<sup>a</sup>, foi realizado a montagem do circuito. Como a fonte de alimentação é de 9v, foi utilizado um regulador de tensão LM7805 para conversão da tensão em 5v, para assim alimentar o Microcontrolador PIC18F2450 e o Servomotor que tem sua tensão de trabalho em 5v.

O servomotor foi conectado ao PIC através do pino 13, que é uma porta de input/output digital e com função PWM. A alimentação do servo foi feita através de uma ligação direta ao circuito regulador de tensão evitando interferências no funcionamento do microcontrolador. As placas Fotovoltaicas que coleta informações sobre a posição do sol baseado na tensão gerada, foi conectado no pino 2 de entrada analógica. A Figura 5.4 ilustra o circuito montado na protoboard.





O código foi escrito obedecendo ao diagrama de fluxo mostrado na Figura 5.5. Ao iniciar a execução do código, é zerado os valores das variáveis do programa, onde é feito o armazenamento dos valores captados pelas placas fotovoltaicas para realizar os comandos de comparação e execução do programa. A leitura da tensão produzida pelas placas é realizada pela porta analógica do pino 2.

Após leitura são obtidos valores que podem variar entre 0 e 1023, de acordo com a tensão sobre as placas fotovoltaicas. A relação entre a tensão e o valor lido na porta analógica pode ser obtida pela fórmula (7).

$$\text{Valor de leitura} = \frac{\text{tensão (v)}}{5.0(v). 1023}$$

(7)

Após a inicialização das variáveis é realizada a primeira leitura dos painéis fotovoltaicos, enquanto esse valor for igual a 0, o programa executa essa função de leitura em frações de tempo, pois esse estado identifica um padrão de não produção de energia o que evita o gasto desnecessário do sistema.

Quando identificado que o painel está gerando energia, o microcontrolador realiza a captação do valor da tensão e faz o armazenamento para posterior comparação com um novo valor captado. O microcontrolador envia um sinal PWM ao servomotor para uma leve movimentação das placas, realiza uma nova captura do valor de tensão e compara com o valor já armazenado. Caso o valor seja maior o microcontrolador repetirá esse ciclo até que identifique uma queda na captação. Após identificar uma queda na tensão produzida pelas placas, o microcontrolador envia um sinal PWM ao servomotor para retornar para a posição anterior, onde foi identificada a maior radiação solar devido a maior tensão gerada pelas placas fotovoltaicas. Esse movimento garante que as placas fiquem sempre perpendiculares à incidência da luz solar tendo sua produção maximizada.

A porta analógica 2 é consultada em frações de tempo, sempre que houver uma queda no valor da tensão com relação ao valor anterior, o

microcontrolador irá enviar um novo sinal PWM para o servomotor e realizar uma nova leitura, repetira esse fluxo até o encontro do ponto máximo.

Ao chegar ao fim do curso do servomotor, o microcontrolador detecta o valor a porta analógica 2 até que o valor seja igual a zero, identificando assim o início da noite e garantindo total produção de energia pelas placas fotovoltaicas. Após detectado o valor igual a zero na porta analógica o microcontrolador envia um sinal PWM ao servomotor para que ele retorne ao ponto inicial, aguardando o amanhecer. Durante esse período o sistema entra em módulo de aguardo por um período de horas determinado.

O código utilizado para a programação do microcontrolador é apresentado em Apêndice "A" deste trabalho.

## 5.5 Fluxograma Geral do Sistema

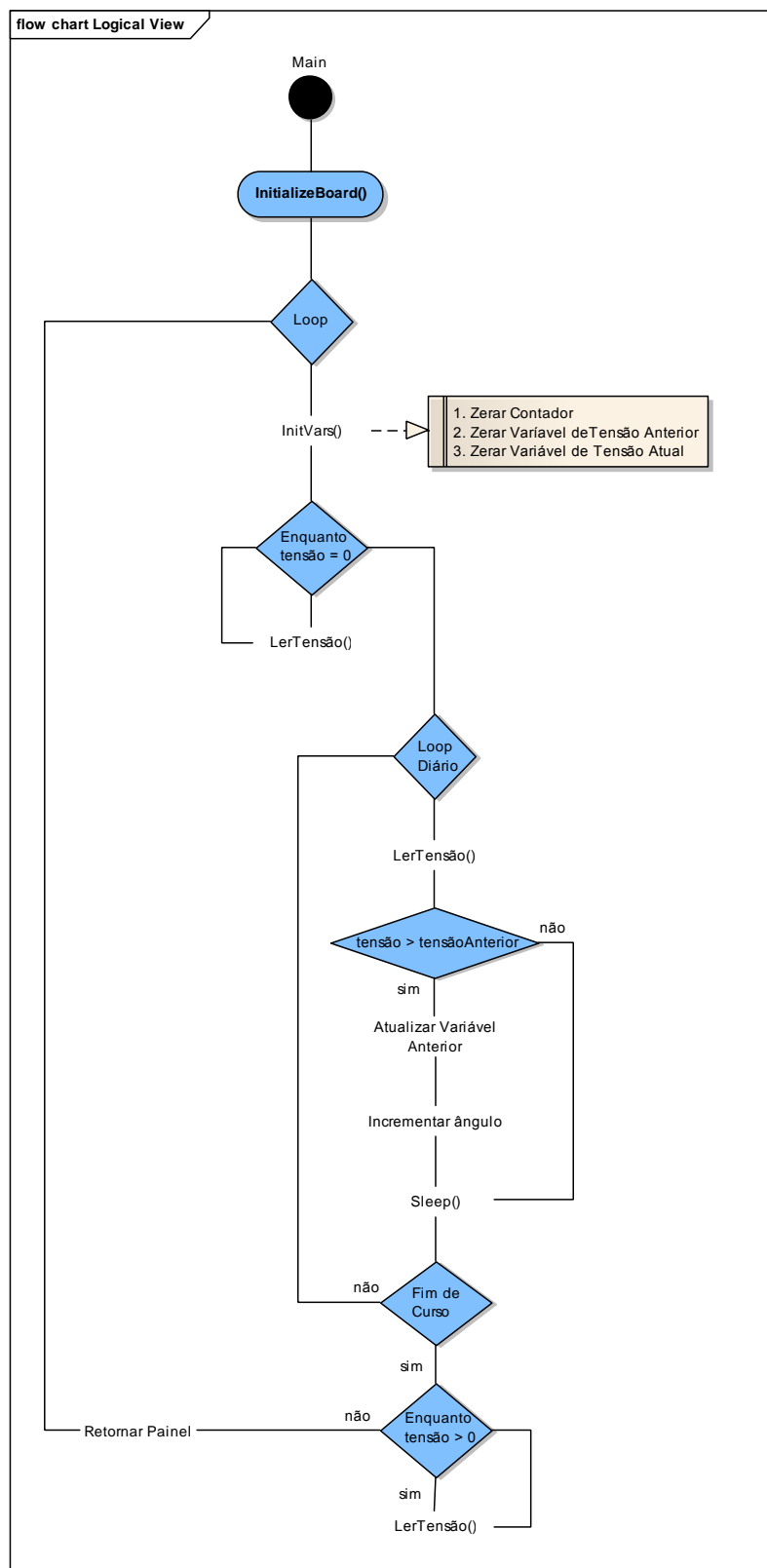
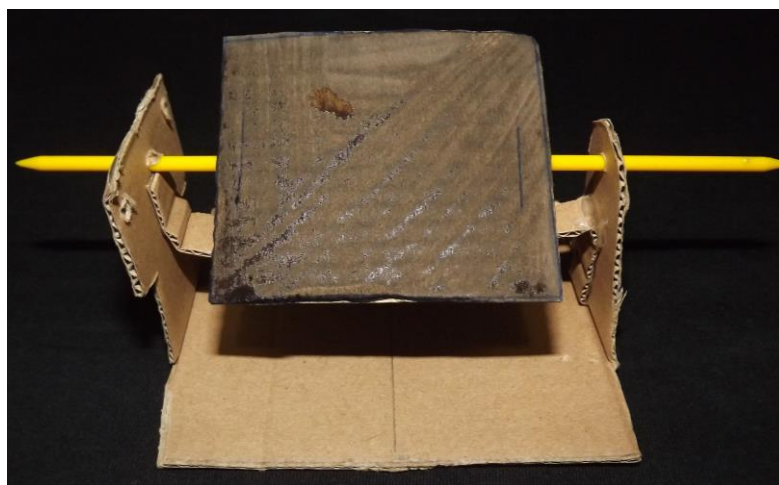


Figura 5.6 – Fluxograma do Programa (Fonte: Autor)

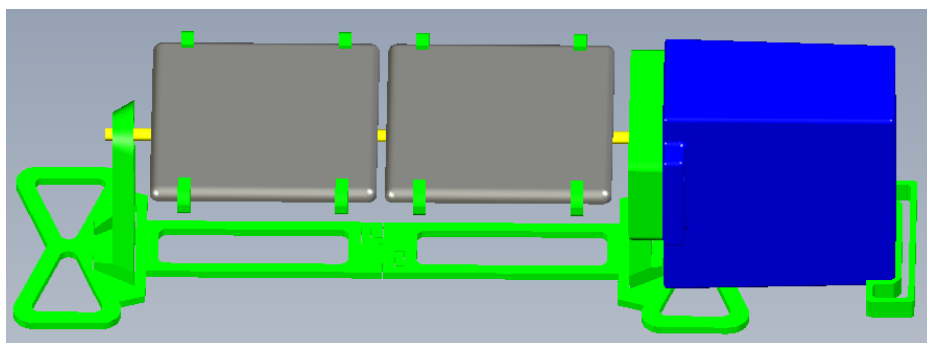
## 5.6 Estrutura do Protótipo

Primeiramente, foi construída uma estrutura de papelão da maquete do protótipo para que fosse identificado o melhor enquadramento das placas fotovoltaicas e do servomotor. A figura 5.7 ilustra o primeiro modelo do protótipo.

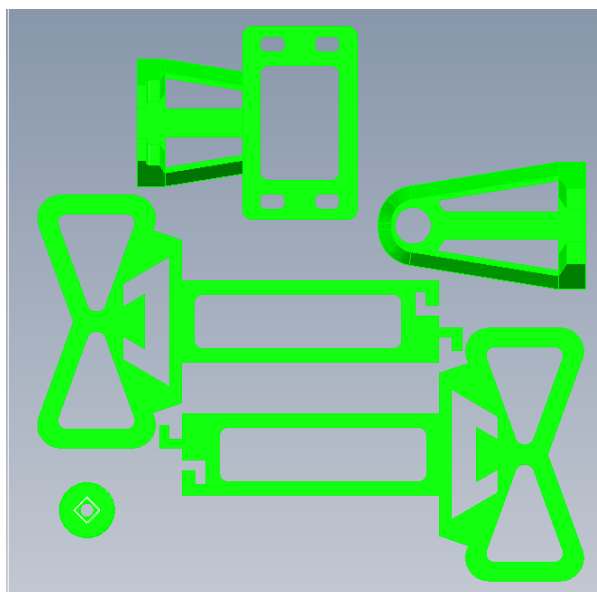


**Figura 5.7 – Primeiro modelo do Protótipo (Fonte: Autor)**

Posterior à definição do layout da estrutura do protótipo, o qual seria construído, foi realizado o desenho no software AutoCAD para que posteriormente fosse enviada a impressora 3D. O layout da estrutura foi criado para suportar as duas placas fotovoltaicas e o servomotor, o qual foi confeccionado em plástico ABS. A posição escolhida para o servomotor foi para anular o uso de engrenagens e o servo controlar diretamente ao eixo de suporte das placas. Dessa forma foi possível ter um controle mais eficiente na movimentação do ângulo das placas. As figuras 5.8 e 5.9 ilustram a projeção do desenho do protótipo no AutoCAD da estrutura do protótipo.

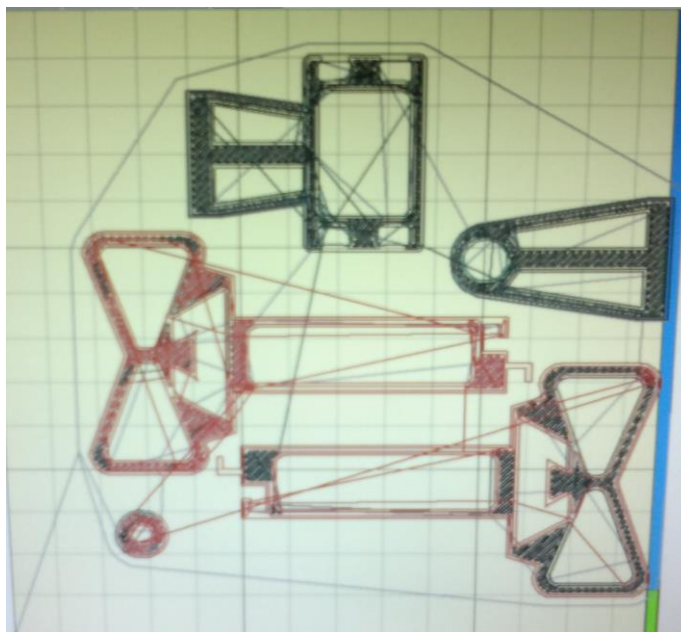


**Figura 5.8 – Projeção do Protótipo no AutoCAD (Fonte: Autor)**



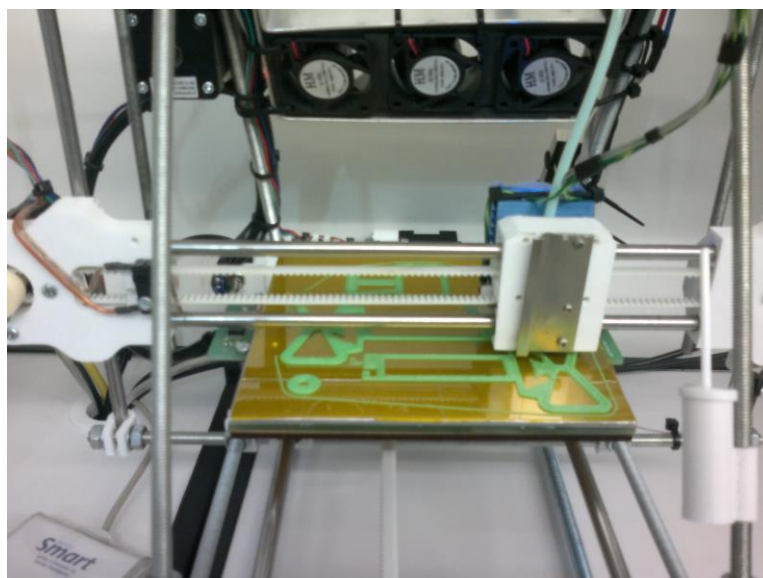
**Figura 5.9 – Projeção do Protótipo desmontado no AutoCAD (Fonte: Autor)**

A figura 5.10 demonstra a projeção do desenho das peças separadas no software da impressora 3D.



**Figura 5.10 – Software Impressora 3D (Fonte: Autor)**

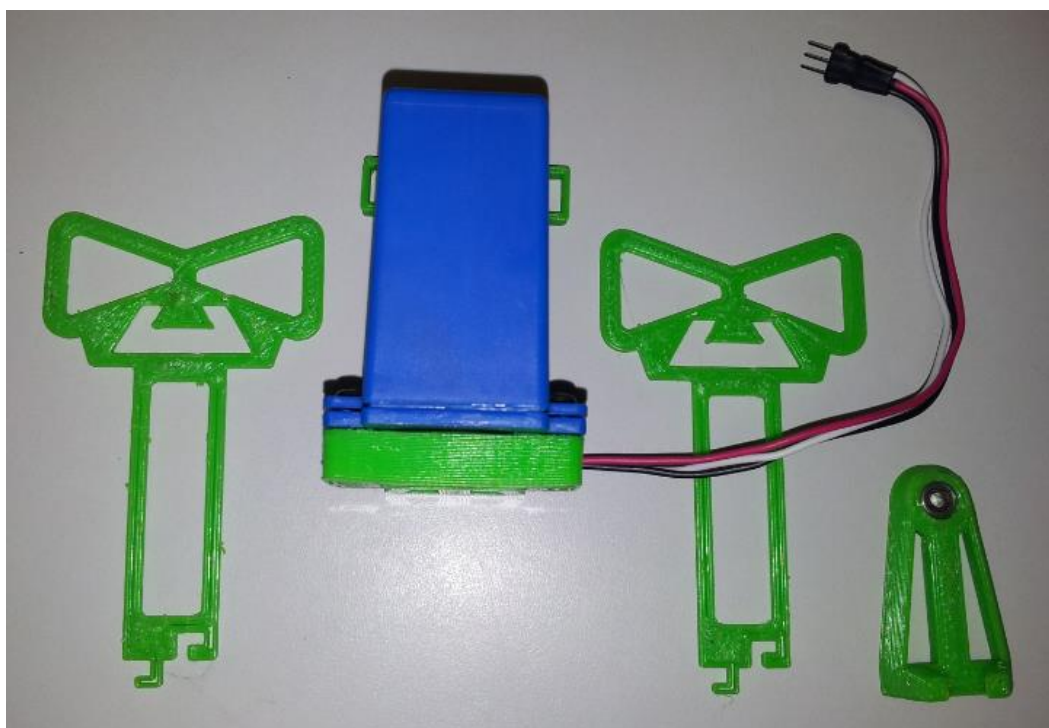
A figura 5.11 ilustra o instante em que era impresso a estrutura do protótipo na impressora 3D. A impressão é realizada com plástico ABS específico para esse tipo de impressora.



**Figura 5.11 – Impressão em Impressora 3D (Fonte: Autor)**

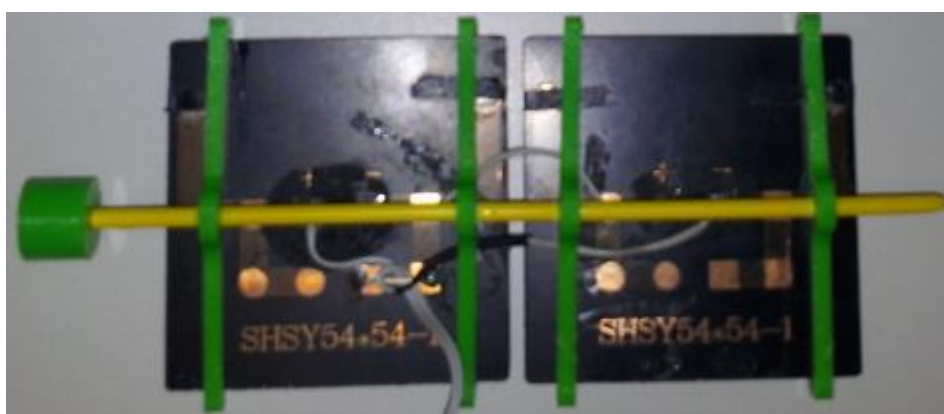


Após a finalização da impressão foi realizado a montagem e acoplamento do motor a estrutura do protótipo, conforme apresentado na figura 5.12.



**Figura 5.12 – Montagem da Estrutura (Fonte: Autor)**

A figura 5.13 demonstra a montagem dos fixadores das placas fotovoltaicas e o eixo que é engatado ao servomotor.

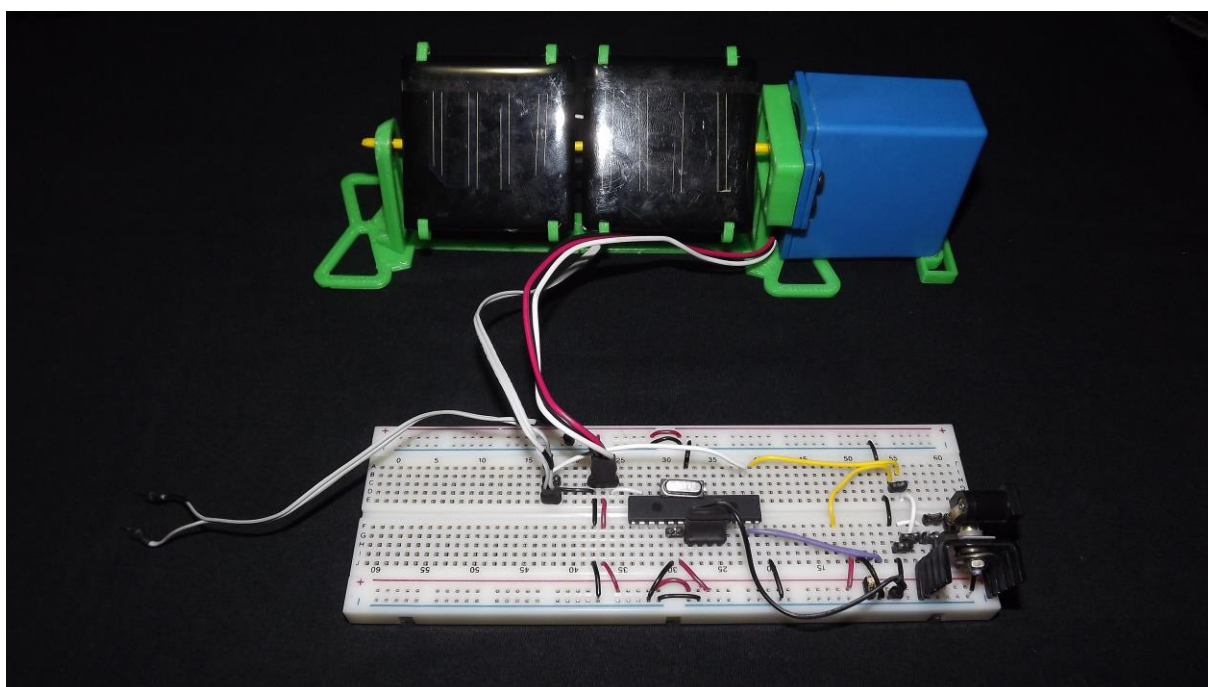


**Figura 5.13 – Suporte das Placas Fotovoltaicas (Fonte: Autor)**



## 5.7 Montagem do Protótipo

Após a montagem da estrutura, foi conectado o servomotor ao circuito do protótipo, conectando o fio de sinal ao pino 2 do micro controlador e os fios de alimentação (positivo e negativo) ao regulador de tensão. Na figura 5.14 é apresentado o protótipo com todas as conexões montadas.



**Figura 5.14 - Protótipo finalizado (Fonte: Autor)**

## **CAPÍTULO 6 – RESULTADOS OBTIDOS**

Neste capítulo é apresentado todo o desenvolvimento das simulações feitas com o projeto, suas funções e todos os problemas ocorridos durante o período de desenvolvimento.

### **6.1 Simulações**

Para comprovação da efetividade do protótipo e sua eficiência, foram realizados duas simulações como testes.

#### **6.1.1 Simulação Seguidor de Intensidade de Luz**

A primeira simulação foi realizada com uma lanterna de super LED para avaliar o funcionamento do protótipo. Como a finalidade era fazer com que o protótipo buscasse sempre a maior incidência da luz e que as placas ficassem sempre em uma posição perpendicular ao ponto emissor da luz, essa simulação ocorreu com sucesso. O Seguidor se manteve sempre voltado perpendicularmente ao ponto de emissão que no caso era a lanterna. Foram realizados vinte repetições para avaliar a precisão do seguidor, em todas elas foram bem sucedidas.

#### **6.1.2 Simulação Comparativa**

Para avaliar um resultado melhor do desempenho do protótipo, foi realizado uma segunda simulação, dessa vez com o intuito de ser um comparativo entre uma placa fixa posicionada em um ângulo de 30° referente ao plano da superfície terrestre, simulando uma instalação convencional, e o protótipo em funcionamento buscando sempre a maior incidência de luz. Ambos foram colocados na posição Leste-Oeste e a fonte de Luz simulava o movimento Solar. Para realizar essa simulação também foi utilizado a lanterna de superLED, um transferidor de 180° e uma régua de 30cm. A lanterna foi fixada na extremidade da régua e a outra

extremidade fixada ao transferidor para medição do ângulo de incidência. Com os resultados obtidos foi possível gerar o quadro 6, onde os resultados foram bem satisfatórios. Os resultados foram obtidos através da media das vinte repetições realizadas com o protótipo.

**Quadro 6 – Comparativo de Tensão produzida**

Posição da Luz	Placa Fixa 30° (Tensão V)	Protótipo (Tensão V)	% de incremento
Leste 15°	2,87	2,64	-8%
Leste 30°	2,91	2,63	-10%
Leste 45°	2,82	2,73	-3%
Leste 60°	2,49	2,9	16%
Leste 75°	2,24	2,74	22%
90°	2,15	2,69	25%
Oeste 75°	2	2,81	41%
Oeste 60°	1,89	2,9	53%
Oeste 45°	0,6	2,77	362%
Oeste 30°	0,15	2,76	1740%
Oeste 15°	0,02	2,8	13900%

**Fonte: Autor**

## 6.2 Problemas Encontrados

Aquisição de componentes eletrônicos em Brasília é bastante limitado. Foi necessário realizar a importação das Placas Fotovoltaicas por questão de custos. Outro fator é o alto preço local, grande parte dos componentes foram adquiridos via internet no estado de São Paulo.

A proposta inicial do Projeto era de trabalhar com o microcontrolador Basicstep da empresa Tato, porém ocorreram falhas na placa gravadora e a aquisição de uma nova placa demoraria bastante.

Durante o período de apresentação do seminário, a placa programadora sofreu um dano e não estava realizando a gravação do Microcontrolador PIC, o que foi necessário a aquisição de uma nova placa.

### 6.3 Orçamento do Projeto

A aquisição dos componentes eletrônicos para o projeto foi realizada e foi fundamental para o projeto. Abaixo o Quadro 6.2 detalha os componentes e o seus devidos custos.

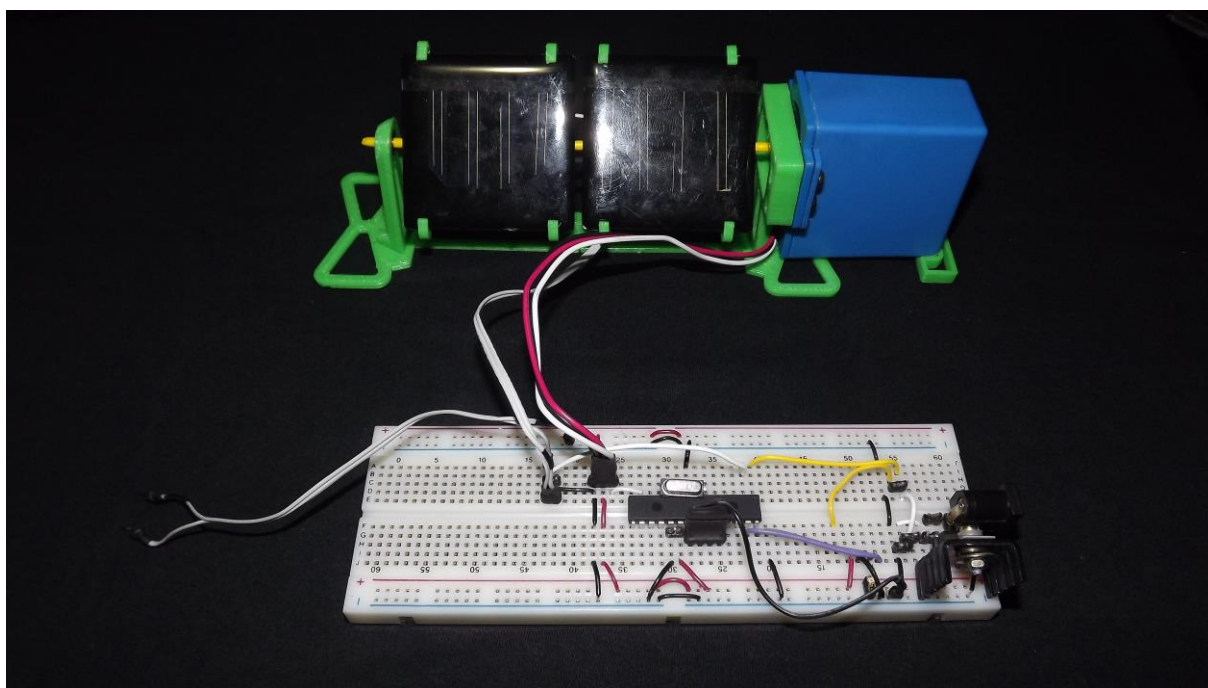
**Quadro 6.1 – Custo do Projeto**

Item	Custo Unitário	Quantidade	Custo Total
PIC18F2450 – MicroChip	R\$ 42,90	1	R\$ 42,90
Cristal Quartzo 10MHz	R\$ 1,40	1	R\$ 1,40
PROTOBOARD MP 830 <sup>a</sup>	R\$ 16,10	1	R\$ 16,10
Placa Fotovoltaica	R\$ 8,00	4	R\$ 24,00
Servo Motor Posição	R\$ 50,00	1	R\$ 50,00
Impressão da Estrutura do Prototipo	R\$ 200,00	1	R\$ 200,00
Regulador de Tensão LM7805	R\$ 1,79	1	R\$ 1,79
Resistor 1K ohms	R\$ 0,30	10	R\$ 3,00
Resistor 10K ohms	R\$ 0,30	5	R\$ 1,50
Resistor 100K ohms	R\$ 0,30	5	R\$ 1,50
Multímetro digital - Minipa ET-880	R\$ 49,00	1	R\$ 49,00
Fonte de Alimentação 9 V	R\$ 25,00	1	R\$ 25,00
<b>TOTAL</b>	-	-	<b>R\$ 416,19</b>

Fonte: Autor.

## 6.4 Protótipo Final

Conforme mostrado na Figura 6.1, o projeto com todos seus componentes conectados.



**Figura 6.1 – Projeto Completo (Fonte: Autor)**

## **CAPÍTULO 7 – CONSIDERAÇÕES FINAIS**

Neste capítulo são apresentadas as conclusões e as propostas para trabalhos futuros, que podem utilizar este projeto como base.

### **7.1 Conclusões**

Foi desenvolvido neste trabalho um dispositivo de maximização de captação da luz solar . Com os testes realizados em situações de simulação do movimento do Sol, o protótipo funcionou conforme o previsto, e desta forma pode ser desenvolvido em escala maior para que seja implantado em ambientes remotos onde há escassez de energia ou até mesmo em instalações de placas Fotovoltaicas fixas, maximizando sua produção.

A proposta de se construir um seguidor solar o qual realizou a movimentação da placa fotovoltaica, mantendo-a sempre perpendicular aos raios solares e no maior ponto de produção da placa foram alcançados, o protótipo funciona conforme o planejado. Ocorreram alguns imprevistos durante o processo de confecção, mas isso foi superado.

O microcontrolador PIC18F2450 atendeu completamente as funções do protótipo, a biblioteca do PIC18 voltada para o uso da função PWM, facilitou bastante o desenvolvimento da logica do microcontrolador. Como o PIC 18F possui bastantes funcionalidades e dispõem de vários pinos, ainda poderiam ser adicionados outros componentes caso futuramente seja incluído ao projeto. O circuito foi projetado com sucesso e o funcionamento do mesmo foi excelente atendendo as expectativas criadas no inicio do projeto. A integração entre as placasfotovoltaicas, servomotor e o microcontrolador funcionaram conforme o planejado.

O Protótipo foi construído em escala reduzida para comprovação de sua eficiência. Para uma construção em escala real, deverá ser alterado o motor que realiza o deslocamento das placas para um motor de indução, para que possa suportar o peso de placas mais robustas e potentes. Para a estrutura de sustentação deverá ser construído e, material mais resistente como aço ou alumínio, que suporte

todo o peso de motor, placas e eixos. O sistema microcontrolado poderá ser mantido conforme construído para o protótipo, mantendo-se também a lógica de programação.

## **7.2 Propostas para Trabalhos Futuros**

Com o desenvolvimento da monografia e do modelo apresentado, algumas sugestões para evolução do estudo foram visualizadas, são elas:

- Incluir na estrutura e na lógica de programação o deslocamento norte-sul, fazendo com que a placa realize o movimento completo do sol;
- Alterar a estrutura do protótipo para adaptar o servomotor de uma forma que alterando o motor não impactará o mecanismo
- Adicionar ao protótipo um sistema de baterias, carregadas pela energia gerada pelas placas fotovoltaicas transformando em um sistema independente de fontes externas.
- Foram realizados vários testes comprovando a eficiência do protótipo em relação à uma instalação fixa, deve-se realizar testes que certifique essa eficiência conforme padrões oficiais de certificações.

## REFERÊNCIAS BIBLIOGRÁFICAS

CRESESB, 2006 (Centro de Referência para Energia Solar e Eólica Sérgio de Salvo Brito), Manual de Engenharia para Sistemas Fotovoltaicos.

GREENPRO. Energia Fotovoltaica. Disponível em: <<http://www.greenpro.de/po/fotovoltaico.pdf>>. Acessado em 08 de mai. 2013.

Heliotec. Geometria Solar. Disponível em: <[http://www.heliodon.com.br/geometria\\_solar.html](http://www.heliodon.com.br/geometria_solar.html)>. Acessado em 13 de mar. 2013.

MICROCHIP TECHNOLOGY INC. PIC18F2450. DataSheet: 28/40/44-Pin Enhanced Flash Microcontrollers. 2006. Disponível em: <<http://ww1.microchip.com/downloads/en/DeviceDoc/39760a.pdf>> Acesso em 2 mai. 2013.

Motions of the Sun. Motions of the Sun Simulator. Disponível em: <<http://astro.unl.edu/classaction/animations/coordsmotion/sunmotions.html>>. Acessado em 08 de mai. 2013.

NASCIMENTO, Cássio Araújo do; Células Fotovoltaicas. Departamento de Engenharia da Universidade Federal de Lavras, 2004.

O Globo. Um milhão de lares brasileiros não têm energia elétrica. Publicado em 26 de dez 2012. Disponível em <<http://oglobo.globo.com/economia/um-milhao-de-lares-brasileiros-nao-tem-energia-eletrica-7132890>>. Acesso em 17 de mar 2013.

PEREIRA, F. Microcontroladores PIC: Programação em C. 7ª. Ed. São Paulo: Érica, 2007.



SEESP. Debate de impactos ambientais da geração de energia elétrica. Disponível em: <<http://www.seesp.org.br/site/imprensa/noticias/item/4106-unesp-debate-impactos-ambientais-da-gera%C3%A7%C3%A3o-de-energia-el%C3%A9trica.html>>. Acessado em 08 de mai. 2013.

SOUZA, David Jose de; SOUSA, Daniel Rodrigues de. Desbravando o Microcontroladores PIC18: Ensino Didático. São Paulo: Érica, 2012.

SOUZA, David Jose de; SOUSA, Daniel Rodrigues de; LAVINIA, Nicolás César. Desbravando o Microcontroladores PIC18: Recursos Avançados. São Paulo: Érica, 2012.

SOUZA, D. J. de; LAVINIA, N. C. Conectando o PIC 16F877A: Recursos Avançados. 2ª. Ed. São Paulo: Érica, 2005.

TIPLER, Paul A, Física, Eletricidade, Magnetismo e Ótica, Vol 2.

## APÊNDICE A – CÓDIGO FONTE DO DISPOSITIVO

```
#include "Main.h"
#include "fgdelay.h"

#pragma config PLLDIV=1, CPUDIV=OSC3_PLL4, USBDIV=2, FOSC=XT_XT,
BOR=OFF, WDT=OFF, XINST=OFF, PBDEN=OFF, MCLRE=OFF, STVREN=OFF,
LVP=OFF

/**
 * Configuração do clock
 *
 * PLLDIV = 1      No prescale (4 MHz oscillator input drives PLL directly)
 * CPUDIV = OSC3_PLL4 [Primary Oscillator Src: /3][96 MHz PLL Src: /4]
 *
 * Com essa configuração o PIC irá operar em 1,3333 MHz (4MHz/3)
 */

static void InitializeBoard(void);

#include "adc.h"

#define SERVO_POS_INICIAL    95
#define SERVO_POS_FINAL      170

void main(void) {

    UINT8 i, servoPosicao, pontoMaximo;
    UINT16 adcRead, adcLast;
    BOOL fimDoDia = FALSE;

    InitializeBoard();

    LED1 = 0;
    LED2 = 0;
    LED3 = 0;
    LED4 = 0;

    while (1) {

        // Posição inicial do servo
        servoPosicao = SERVO_POS_INICIAL;

        startPWM(servoPosicao);
        delaySegundos(1);

        adcRead = 0;
        adcLast = 0;
        pontoMaximo = 0;
    }
}
```

```

while (1) {

    LED4 = 1;

    readANS(&adcRead, 0, 10);

    if(fimDoDia) {
        if(adcRead < 10) {
            fimDoDia = FALSE;
            LED4 = 0;
            delaySegundos(5);
            break;
        }
    }
    else {

        if(adcRead == adcLast) {
            i--;
            continue;
        }

        if(adcRead > adcLast) {
            if(servoPosicao < SERVO_POS_FINAL)

++servoPosicao;

        }
        else {
            if(servoPosicao+2 > pontoMaximo) --servoPosicao;
            delay_ms(100);
        }

        if(pontoMaximo < servoPosicao) pontoMaximo = servoPosicao;

        startPWM(servoPosicao);
        delay_ms(100);
        stopPWM();
        delay_ms(100);

        adcLast = adcRead;

        if(servoPosicao == SERVO_POS_FINAL) {
            fimDoDia = TRUE;
        }
    }
}

/**
 * Faz a leitura de um canal do conversor A/D
 */
void readANS(UINT16 *data, UINT8 canal, UINT8 size) {

```

```

    UINT8 i;
    UINT32 vetor = 0;
    SetChanADC(canal);
    for(i=0; i<size; i++) {
        ConvertADC();
        while( BusyADC() );
        vetor += (UINT32)ReadADC();
    }
    *data = round((float)vetor / size);
}

// Funções de utilidade geral
UINT16 round (float x) {
    return (UINT16)(x + 0.5f);
}

static void InitializeBoard(void) {

    TRIS_LED1 = 0;
    TRIS_LED2 = 0;
    TRIS_LED3 = 0;
    TRIS_LED4 = 0;

    TRIS_ADC = 1;

    // Config A/D
    OpenADC( ADC_RIGHT_JUST & ADC_FOSC_2 & ADC_2_TAD,
            ADC_CH0 & ADC_INT_OFF & ADC_VREFPLUS_VDD &
ADC_REF_VDD_VSS,
            ADC_0ANA);
}

```

## APÊNDICE B – BIBLIOTECA PWM

```

#ifndef __PWM_H
#define __PWM_H
/*****
***
// *          PULSE WIDTH MODULATION (PWM)  PERIPHERAL LIBRARY
HEADER FILE

****
***
* FileName:          pwm.h
* Dependencies:      See include below
* Processor:         PIC18
* Compiler:          MCC18
* Company:           Microchip Technology, Inc.
*
* Software License Agreement
* The software supplied herewith by Microchip Technology Incorporated
* (the "Company") for its PICmicro® Microcontroller is intended and
* supplied to you, the Company's customer, for use solely and
* exclusively on Microchip PICmicro Microcontroller products. The
* software is owned by the Company and/or its supplier, and is
* protected under applicable copyright laws. All rights are reserved.
* Any use in violation of the foregoing restrictions may subject the
* user to criminal sanctions under applicable laws, as well as to
* civil liability for the breach of the terms and conditions of this
* license.
*
* THIS SOFTWARE IS PROVIDED IN AN "AS IS" CONDITION. NO WARRANTIES,
* WHETHER EXPRESS, IMPLIED OR STATUTORY, INCLUDING, BUT NOT
LIMITED
* TO, IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A
* PARTICULAR PURPOSE APPLY TO THIS SOFTWARE. THE COMPANY SHALL
NOT,
* IN ANY CIRCUMSTANCES, BE LIABLE FOR SPECIAL, INCIDENTAL OR
* CONSEQUENTIAL DAMAGES, FOR ANY REASON WHATSOEVER.

****
**/

#include <pconfig.h>

/* There are five library modules, corresponding to register names:
* PWM1 (CCPCON1)
* PWM2 (CCPCON2)
* PWM3 (CCPCON3)
* PWM4 (CCPCON4)
* PWM5 (CCPCON5)
* EPWM1 (ECCPCON1)

```

```

* Each module is defined only for those devices for which the register
* name is defined.
* Note that the 'E' prefix indicates enhanced CCP capability (in ECCPCON1,
* for example); however, the absence of the 'E' prefix does not imply
* lack of enhanced capability. The usage of the 'E' prefix is indicative
* of the register naming convention in the datasheets, not the
* functionality.
*/

```

```

/* For each module, there are three routines: an 'open' routine,
* a 'SetDC' routine, and a 'close' routine.
*
* The 'open' routine
* - sets the PWM mode as single output;
* - sets the PWM period, as specified by the routine's parameter;
* - configures the CCPx port pin for output;
* - begins the PWM operation.
*
* The 'SetDC' routine sets the duty cycle, as specified by the routine's
* parameter.
*
* The 'close' routine turns off PWM operation and configures the CCPx port
* pin to input.
*
* For devices with enhanced CCP capability, an additional 'SetOutput'
* routine is provided. This routine takes two parameters:
* - 'output_config' is the output configuration:
*   single output
*   full bridge output forward
*   half bridge output
*   full bridge output reverse
* - 'pwm_mode' is the PWM mode:
*   PxA,PxC active high; PxB,PxD active high
*   PxA,PxC active high; PxB,PxD active low
*   PxA,PxC active low; PxB,PxD active high
*   PxA,PxC active low; PxB,PxD active low
* These parameters are represented by symbolic constants defined below.
* The 'SetOutput' routines set the output configuration and PWM mode;
* they also configure the port pins corresponding to the output
* configuration for output. Note that unlike the 'open' routines, these
* routines do not set the period nor begin the PWM operation.
*/

```

```

/* Union used to hold the 10-bit duty cycle */
union PWMDC
{
    unsigned int lpwm;
    char bpwm[2];
};

```

```

/* ***** Enhanced CCP ***** */
/* These devices have enhanced CCP capability. For PWM, this means up to
 * four outputs. The 'SetOutput' routines are defined for those modules
 * with this enhanced functionality. These routines set the output
 * configuration and the PWM mode, as well as configure the appropriate
 * port pins for output. Note that the port pins configured are those
 * corresponding to microcontroller mode.
 */
#define PWM_V4

#if defined (PWM_V4) || defined (PWM_V5) || defined (PWM_V6) || \
    defined (EPWM_V7) || defined (PWM_V8) || defined (PWM_V10) || \
    defined (PWM_V11) || defined (PWM_V12) || defined (PWM_V13) || \
    defined (PWM_V14) || defined (PWM_V14_1) || defined (EPWM_V14) \
    || defined (EPWM_V14_1) || defined (EPWM_V14_2) || defined (PWM_V14_2)

#ifndef USE_OR_MASKS

#define SINGLE_OUT          0b00111111 /*Single output: P1A modulated; P1B, P1C,
P1D assigned as port pins*/
#define FULL_OUT_FWD        0b01111111 /*Full-bridge output forward: P1D
modulated; P1A active; P1B, P1C inactive*/
#define HALF_OUT            0b10111111 /*Half-bridge output: P1A, P1B modulated
with dead-band control; P1C, P1D assigned as port pins*/
#define FULL_OUT_REV        0b11111111 /*Full-bridge output reverse: P1B
modulated; P1C active; P1A, P1D inactive*/

#else

#define SINGLE_OUT          0b00000000 /*Single output: P1A modulated; P1B, P1C,
P1D assigned as port pins*/
#define FULL_OUT_FWD        0b01000000 /*Full-bridge output forward: P1D
modulated; P1A active; P1B, P1C inactive*/
#define HALF_OUT            0b10000000 /*Half-bridge output: P1A, P1B modulated
with dead-band control; P1C, P1D assigned as port pins*/
#define FULL_OUT_REV        0b11000000 /*Full-bridge output reverse: P1B
modulated; P1C active; P1A, P1D inactive*/
#define PWM_OP_MODE_MASK (~FULL_OUT_REV)//Mask      PWM      Output
Configuration bits

#endif

#define IS_DUAL_PWM(config) ((config) == HALF_OUT)

#define IS_QUAD_PWM(config) ((config) == FULL_OUT_FWD || \
    (config) == FULL_OUT_REV)

/* The PWM mode is passed as the second parameter to the
 * 'SetOutput' routines. */

```

```
#ifndef USE_OR_MASKS
```

```
#define PWM_MODE_1 0b11111100 /* PxA,PxC active high, PxB,PxD active high */
#define PWM_MODE_2 0b11111101 /* PxA,PxC active high, PxB,PxD active low */
#define PWM_MODE_3 0b11111110 /* PxA,PxC active low, PxB,PxD active high */
#define PWM_MODE_4 0b11111111 /* PxA,PxC active low, PxB,PxD active low */
```

```
#else
```

```
#define PWM_MODE_1 0b00001100 /* PxA,PxC active high, PxB,PxD active high */
#define PWM_MODE_2 0b00001101 /* PxA,PxC active high, PxB,PxD active low */
#define PWM_MODE_3 0b00001110 /* PxA,PxC active low, PxB,PxD active high */
#define PWM_MODE_4 0b00001111 /* PxA,PxC active low, PxB,PxD active low */
#define PWM_MODE_MASK (~PWM_MODE_4) //Mask PWM Output
Configuration bits
```

```
#endif
```

```
#endif
```

```
#if defined (PWM_V14) || defined (PWM_V14_1)
```

```
#define __CONFIG3H 0x300005
```

```
#ifndef USE_OR_MASKS
```

```
//***** CCP source - TIMER selection
*****
```

```
//*****CCP4*****
```

```
#define CCP_4_SEL_TMR12 0b11001111 //CCP selects TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_4_SEL_TMR34 0b11011111 //CCP selects TIMER3 for
Capture & Compare and TIMER4 for PWM
#define CCP_4_SEL_TMR36 0b11101111 //CCP selects TIMER3 for
Capture & Compare and TIMER6 for PWM
```

```
//*****CCP5*****
```

```
#define CCP_5_SEL_TMR12 0b11001111 //CCP selects TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_5_SEL_TMR54 0b11011111 //CCP selects TIMER5 for
Capture & Compare and TIMER4 for PWM
```

```
//*****CCP6*****
```

```
#define CCP_6_SEL_TMR12 0b11001111 //CCP selects TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_6_SEL_TMR52 0b11011111 //CCP selects TIMER5 for
Capture & Compare and TIMER2 for PWM
```

```
//*****CCP7*****
```



```

#define CCP_7_SEL_TMR12          0b11001111    //CCP selects TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_7_SEL_TMR54          0b11011111    //CCP selects TIMER5 for
Capture & Compare and TIMER4 for PWM
#define CCP_7_SEL_TMR56          0b11101111    //CCP selects TIMER5 for
Capture & Compare and TIMER6 for PWM
#define CCP_7_SEL_TMR58          0b11111111    //CCP selects TIMER5 for
Capture & Compare and TIMER8 for PWM

#if defined (PWM_V14)
//*****CCP8*****
#define CCP_8_SEL_TMR12          0b11001111    //CCP selects TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_8_SEL_TMR74          0b11011111    //CCP selects TIMER7 for
Capture & Compare and TIMER4 for PWM
#define CCP_8_SEL_TMR76          0b11101111    //CCP selects TIMER7 for
Capture & Compare and TIMER6 for PWM
#elif defined (PWM_V14_1)
//*****CCP8*****
#define CCP_8_SEL_TMR12          0b11001111    //CCP selects TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_8_SEL_TMR14          0b11011111    //CCP selects TIMER1 for
Capture & Compare and TIMER4 for PWM
#define CCP_8_SEL_TMR16          0b11101111    //CCP selects TIMER1 for
Capture & Compare and TIMER6 for PWM
#endif

#if defined (PWM_V14)
//*****CCP9*****
#define CCP_9_SEL_TMR12          0b11001111    //CCP selects TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_9_SEL_TMR74          0b11011111    //CCP selects TIMER7 for
Capture & Compare and TIMER4 for PWM

//*****CCP10*****
#define CCP_10_SEL_TMR12         0b11001111    //CCP selects TIMER1 for Capture
& Compare and TIMER2 for PWM
#define CCP_10_SEL_TMR72         0b11011111    //CCP selects TIMER7 for Capture
& Compare and TIMER2 for PWM
#endif

#else //USE_OR_MASKS

//*****          CCP          source          -          TIMER          selection
*****
//*****CCP4*****
#define CCP_4_SEL_TMR12          0b00000000    //CCP selects TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_4_SEL_TMR34          0b00010000    //CCP selects TIMER3 for
Capture & Compare and TIMER4 for PWM

```

```

#define CCP_4_SEL_TMR36          0b00100000    //CCP selects TIMER3 for
Capture & Compare and TIMER6 for PWM
#define CCP_4_SEL_TMR_MASK      (~0b00110000)    //Maks    CCP    TIMER
Source selection bits

//*****CCP5*****
#define CCP_5_SEL_TMR12          0b00000000    //CCP selects TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_5_SEL_TMR54          0b00010000    //CCP selects TIMER5 for
Capture & Compare and TIMER4 for PWM
#define CCP_5_SEL_TMR_MASK      (~0b00010000)    //Maks    CCP    TIMER
Source selection bits

//*****CCP6*****
#define CCP_6_SEL_TMR12          0b00000000    //CCP selects TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_6_SEL_TMR52          0b00010000    //CCP selects TIMER5 for
Capture & Compare and TIMER2 for PWM
#define CCP_6_SEL_TMR_MASK      (~0b00010000)    //Maks    CCP    TIMER
Source selection bits

//*****CCP7*****
#define CCP_7_SEL_TMR12          0b00000000    //CCP selects TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_7_SEL_TMR54          0b00010000    //CCP selects TIMER5 for
Capture & Compare and TIMER4 for PWM
#define CCP_7_SEL_TMR56          0b00100000    //CCP selects TIMER5 for
Capture & Compare and TIMER6 for PWM
#define CCP_7_SEL_TMR58          0b00110000    //CCP selects TIMER5 for
Capture & Compare and TIMER8 for PWM
#define CCP_7_SEL_TMR_MASK      (~0b00110000)    //Maks    CCP    TIMER
Source selection bits

#if defined (PWM_V14)
//*****CCP8*****
#define CCP_8_SEL_TMR12          0b00000000    //CCP selects TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_8_SEL_TMR74          0b00010000    //CCP selects TIMER7 for
Capture & Compare and TIMER4 for PWM
#define CCP_8_SEL_TMR76          0b00100000    //CCP selects TIMER7 for
Capture & Compare and TIMER6 for PWM
#define CCP_8_SEL_TMR_MASK      (~0b00110000)    //Maks    CCP    TIMER
Source selection bits
#elif defined (PWM_V14_1)
//*****CCP8*****
#define CCP_8_SEL_TMR12          0b00000000    //CCP selects TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_8_SEL_TMR14          0b00010000    //CCP selects TIMER1 for
Capture & Compare and TIMER4 for PWM
#define CCP_8_SEL_TMR16          0b00100000    //CCP selects TIMER1 for
Capture & Compare and TIMER6 for PWM

```

```

#define CCP_8_SEL_TMR_MASK      (~0b00110000)      //Maks    CCP    TIMER
Source selection bits
#endif

#if defined (PWM_V14)
//*****CCP9*****
#define CCP_9_SEL_TMR12          0b00000000          //CCP selects TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_9_SEL_TMR74          0b00010000          //CCP selects TIMER7 for
Capture & Compare and TIMER4 for PWM
#define CCP_9_SEL_TMR_MASK      (~0b00010000)      //Maks    CCP    TIMER
Source selection bits

//*****CCP10*****
#define CCP_10_SEL_TMR12         0b00000000 //CCP selects TIMER1 for Capture
& Compare and TIMER2 for PWM
#define CCP_10_SEL_TMR72         0b00010000 //CCP selects TIMER7 for Capture
& Compare and TIMER2 for PWM
#define CCP_10_SEL_TMR_MASK      (~0b00010000)      //Maks          CCP
TIMER Source selection bits
#endif

#endif //USE_OR_MASKS
#endif //main version

#if defined (EPWM_V14) || defined (EPWM_V14_1)
#ifndef USE_OR_MASKS
//*****          ECCP          source          -          TIMER          selection
*****
//*****ECCP1*****
#define ECCP_1_SEL_TMR12          0b10001111          //ECCP selects  TIMER1
for Capture & Compare and TIMER2 for PWM
#define ECCP_1_SEL_TMR34          0b10011111          //ECCP selects  TIMER3
for Capture & Compare and TIMER4 for PWM
#define ECCP_1_SEL_TMR36          0b10101111          //ECCP selects  TIMER3
for Capture & Compare and TIMER6 for PWM
#define ECCP_1_SEL_TMR38          0b10111111          //ECCP selects  TIMER3
for Capture & Compare and TIMER8 for PWM
#if defined (EPWM_V14)
#define ECCP_1_SEL_TMR310         0b11001111          //ECCP selects  TIMER3
for Capture & Compare and TIMER10 for PWM
#define ECCP_1_SEL_TMR312         0b11011111          //ECCP selects  TIMER3
for Capture & Compare and TIMER12 for PWM
#endif
#endif

//*****ECCP2*****
#define ECCP_2_SEL_TMR12          0b10001111          //ECCP selects  TIMER1
for Capture & Compare and TIMER2 for PWM
#define ECCP_2_SEL_TMR34          0b10011111          //ECCP selects  TIMER3
for Capture & Compare and TIMER4 for PWM

```

```

#define ECCP_2_SEL_TMR36          0b10101111    //ECCP selects TIMER3
for Capture & Compare and TIMER6 for PWM
#define ECCP_2_SEL_TMR38          0b10111111    //ECCP selects TIMER3
for Capture & Compare and TIMER8 for PWM
#if defined (EPWM_V14)
#define ECCP_2_SEL_TMR310         0b11001111    //ECCP selects TIMER3
for Capture & Compare and TIMER10 for PWM
#endif

/*****ECCP3*****/
#define ECCP_3_SEL_TMR12          0b11001111    //ECCP selects TIMER1
for Capture & Compare and TIMER2 for PWM
#define ECCP_3_SEL_TMR34          0b11011111    //ECCP selects TIMER3
for Capture & Compare and TIMER4 for PWM
#define ECCP_3_SEL_TMR36          0b11101111    //ECCP selects TIMER3
for Capture & Compare and TIMER6 for PWM
#define ECCP_3_SEL_TMR38          0b11111111    //ECCP selects TIMER3
for Capture & Compare and TIMER8 for PWM

#else //USE_OR_MASKS

/***** ECCP source - TIMER selection *****/
/*****ECCP1*****/
#define ECCP_1_SEL_TMR12          0b00000000    //ECCP selects TIMER1
for Capture & Compare and TIMER2 for PWM
#define ECCP_1_SEL_TMR34          0b00010000    //ECCP selects TIMER3
for Capture & Compare and TIMER4 for PWM
#define ECCP_1_SEL_TMR36          0b00100000    //ECCP selects TIMER3
for Capture & Compare and TIMER6 for PWM
#define ECCP_1_SEL_TMR38          0b00110000    //ECCP selects TIMER3
for Capture & Compare and TIMER8 for PWM
#if defined (EPWM_V14)
#define ECCP_1_SEL_TMR310         0b01000000    //ECCP selects TIMER3
for Capture & Compare and TIMER10 for PWM
#define ECCP_1_SEL_TMR312         0b01010000    //ECCP selects TIMER3
for Capture & Compare and TIMER12 for PWM
#endif
#define ECCP_1_SEL_TMR_MASK      (~0b01110000)    //Maks
ECCP TIMER Source selection bits

/*****ECCP2*****/
#define ECCP_2_SEL_TMR12          0b00000000    //ECCP selects TIMER1
for Capture & Compare and TIMER2 for PWM
#define ECCP_2_SEL_TMR34          0b00010000    //ECCP selects TIMER3
for Capture & Compare and TIMER4 for PWM
#define ECCP_2_SEL_TMR36          0b00100000    //ECCP selects TIMER3
for Capture & Compare and TIMER6 for PWM
#define ECCP_2_SEL_TMR38          0b00110000    //ECCP selects TIMER3
for Capture & Compare and TIMER8 for PWM

```

```

#if defined (EPWM_V14)
#define ECCP_2_SEL_TMR310          0b01000000    //ECCP  selects  TIMER3
for Capture & Compare and TIMER10 for PWM
#endif
#define ECCP_2_SEL_TMR_MASK        (~0b01110000)    //Maks
ECCP TIMER Source selection bits

//*****ECCP3*****
#define ECCP_3_SEL_TMR12           0b00000000    //ECCP  selects  TIMER1
for Capture & Compare and TIMER2 for PWM
#define ECCP_3_SEL_TMR34           0b00010000    //ECCP  selects  TIMER3
for Capture & Compare and TIMER4 for PWM
#define ECCP_3_SEL_TMR36           0b00100000    //ECCP  selects  TIMER3
for Capture & Compare and TIMER6 for PWM
#define ECCP_3_SEL_TMR38           0b00110000    //ECCP  selects  TIMER3
for Capture & Compare and TIMER8 for PWM
#define ECCP_3_SEL_TMR_MASK        (~0b00110000)    //Maks
ECCP TIMER Source selection bits

#endif //USE_OR_MASKS
#endif //main version

#if defined (PWM_V14_2) || defined (EPWM_V14_2)
#ifndef USE_OR_MASKS
#define ECCP_1_SEL_TMR12           0b11101111    //ECCP  selects  TIMER1   for
Capture & Compare and TIMER2 for PWM
#define ECCP_1_SEL_TMR34           0b11111111    //ECCP  selects  TIMER3   for
Capture & Compare and TIMER4 for PWM

#define CCP_2_SEL_TMR12            0b11101111    //CCP  selects  TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_2_SEL_TMR34            0b11111111    //CCP  selects  TIMER3 for
Capture & Compare and TIMER4 for PWM

#define CCP_3_SEL_TMR12            0b11101111    //CCP  selects  TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_3_SEL_TMR34            0b11111111    //CCP  selects  TIMER3 for
Capture & Compare and TIMER4 for PWM

#define CCP_4_SEL_TMR12            0b11101111    //CCP  selects  TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_4_SEL_TMR34            0b11111111    //CCP  selects  TIMER3 for
Capture & Compare and TIMER4 for PWM

#define CCP_5_SEL_TMR12            0b11101111    //CCP  selects  TIMER1 for
Capture & Compare and TIMER2 for PWM
#define CCP_5_SEL_TMR34            0b11111111    //CCP  selects  TIMER5 for
Capture & Compare and TIMER4 for PWM

```

```

#else
#define ECCP_1_SEL_TMR12      0b00000000    //ECCP  selects  TIMER1  for
Capture & Compare and TIMER2 for PWM
#define ECCP_1_SEL_TMR34      0b00010000    //ECCP  selects  TIMER3  for
Capture & Compare and TIMER4 for PWM
#define ECCP_1_SEL_TMR_MASK    (~0b00010000)    //Maks      ECCP
TIMER Source selection bits

#define CCP_2_SEL_TMR12      0b00000000    //CCP  selects  TIMER1  for
Capture & Compare and TIMER2 for PWM
#define CCP_2_SEL_TMR34      0b00010000    //CCP  selects  TIMER3  for
Capture & Compare and TIMER4 for PWM
#define CCP_2_SEL_TMR_MASK    (~0b00010000)    //Maks      CCP    TIMER
Source selection bits

#define CCP_3_SEL_TMR12      0b00000000    //CCP  selects  TIMER1  for
Capture & Compare and TIMER2 for PWM
#define CCP_3_SEL_TMR34      0b00010000    //CCP  selects  TIMER3  for
Capture & Compare and TIMER4 for PWM
#define CCP_3_SEL_TMR_MASK    (~0b00010000)    //Maks      CCP    TIMER
Source selection bits

#define CCP_4_SEL_TMR12      0b00000000    //CCP  selects  TIMER1  for
Capture & Compare and TIMER2 for PWM
#define CCP_4_SEL_TMR34      0b00010000    //CCP  selects  TIMER3  for
Capture & Compare and TIMER4 for PWM
#define CCP_4_SEL_TMR_MASK    (~0b00010000)    //Maks      CCP    TIMER
Source selection bits

#define CCP_5_SEL_TMR12      0b00000000    //CCP  selects  TIMER1  for
Capture & Compare and TIMER2 for PWM
#define CCP_5_SEL_TMR34      0b00010000    //CCP  selects  TIMER5  for
Capture & Compare and TIMER4 for PWM
#define CCP_5_SEL_TMR_MASK    (~0b00010000)    //Maks      CCP    TIMER
Source selection bits
#endif
#endif

void OpenPWM1 ( char period);
void SetDCPWM1 ( unsigned int duty_cycle);
#if defined (PWM_V4) || defined (PWM_V5) || defined (PWM_V6) ||\
    defined (EPWM_V7) || defined (PWM_V8) || defined (PWM_V10) || \
    defined (PWM_V11) || defined (PWM_V12)|| defined (PWM_V13)

void SetOutputPWM1 ( unsigned char output_config,
                    unsigned char pwm_mode);
#endif
void ClosePWM1 (void);

```

```
#if defined (PWM_V2) || defined (PWM_V3) || defined (PWM_V4) || \
    defined (PWM_V5) || defined (PWM_V6) || defined (PWM_V9) || \
    defined (PWM_V10) || defined (PWM_V11) || defined (PWM_V13)
```

```
void OpenPWM2 ( char period);
void SetDCPWM2( unsigned int duty_cycle);
```

```
#if defined (PWM_V4)
void SetOutputPWM2 ( unsigned char output_config,
                    unsigned char pwm_mode);
```

```
#endif
void ClosePWM2 (void);
```

```
#endif
```

```
#if defined (PWM_V3) || defined (PWM_V4) || defined (PWM_V9)
```

```
void OpenPWM3 ( char period);
void SetDCPWM3 ( unsigned int duty_cycle);
#if defined (PWM_V4)
void SetOutputPWM3 ( unsigned char output_config,
                    unsigned char pwm_mode);
```

```
#endif
void ClosePWM3 (void);
```

```
#endif
```

```
#if defined (PWM_V4) || defined (PWM_V9)
```

```
void OpenPWM4 ( char period);
void SetDCPWM4 ( unsigned int duty_cycle);
void ClosePWM4 (void);
```

```
void OpenPWM5 ( char period);
void SetDCPWM5 ( unsigned int duty_cycle);
void ClosePWM5 (void);
```

```
#endif
```

```
#if defined (PWM_V14_2)
void OpenPWM2 ( unsigned char period, unsigned char timer_source );
void SetDCPWM2 ( unsigned int duty_cycle);
void ClosePWM2 (void);
```

```
void OpenPWM3 ( unsigned char period, unsigned char timer_source );
void SetDCPWM3 ( unsigned int duty_cycle);
```

```

void ClosePWM3 (void);
#endif

#if defined (PWM_V14) || defined (PWM_V14_1) || defined (PWM_V14_2)

void OpenPWM4 ( unsigned char period, unsigned char timer_source );
void SetDCPWM4 ( unsigned int duty_cycle);
void ClosePWM4 (void);

void OpenPWM5 ( unsigned char period, unsigned char timer_source );
void SetDCPWM5 ( unsigned int duty_cycle);
void ClosePWM5 (void);

#if defined (PWM_V14) || defined (PWM_V14_1)
void OpenPWM6 ( unsigned char period, unsigned char timer_source );
void SetDCPWM6 ( unsigned int duty_cycle);
void ClosePWM6 (void);

void OpenPWM7 ( unsigned char period, unsigned char timer_source );
void SetDCPWM7 ( unsigned int duty_cycle);
void ClosePWM7 (void);

void OpenPWM8 ( unsigned char period, unsigned char timer_source );
void SetDCPWM8 ( unsigned int duty_cycle);
void ClosePWM8 (void);

#if defined (PWM_V14)
void OpenPWM9 ( unsigned char period, unsigned char timer_source );
void SetDCPWM9 ( unsigned int duty_cycle);
void ClosePWM9 (void);

void OpenPWM10 ( unsigned char period, unsigned char timer_source );
void SetDCPWM10 ( unsigned int duty_cycle);
void ClosePWM10 (void);
#endif
#endif

#endif

#if defined (EPWM_V7)

void OpenEPWM1 ( char period);
void SetDCEPWM1 ( unsigned int duty_cycle);
void SetOutputEPWM1 ( unsigned char output_config,
                     unsigned char pwm_mode);
void CloseEPWM1 (void);

#endif

#if defined (EPWM_V14) || defined (EPWM_V14_1) || defined (EPWM_V14_2)

```



```

void OpenEPWM1( unsigned char period, unsigned char timer_source );
void SetDCEPWM1 ( unsigned int duty_cycle);
void SetOutputEPWM1 ( unsigned char output_config,
                    unsigned char pwm_mode);
void CloseEPWM1 (void);

```

```

#if defined (EPWM_V14) || defined (EPWM_V14_1)
void OpenEPWM2( unsigned char period, unsigned char timer_source );
void SetDCEPWM2 ( unsigned int duty_cycle);
void SetOutputEPWM2 ( unsigned char output_config,
                    unsigned char pwm_mode);
void CloseEPWM2 (void);

```

```

void OpenEPWM3( unsigned char period, unsigned char timer_source );
void SetDCEPWM3 ( unsigned int duty_cycle);
void SetOutputEPWM3 ( unsigned char output_config,
                    unsigned char pwm_mode);
void CloseEPWM3 (void);
#endif
#endif

```

```

#if defined CC4_IO_V1
#define PWM4_TRIS TRISDbits.TRISD2
#elif defined CC4_IO_V2
#define PWM4_TRIS TRISBbits.TRISB4
#elif defined CC8_IO_V1
#define PWM4_TRIS TRISCbits.TRISC7
#elif defined CC8_IO_V2
#define PWM4_TRIS TRISCbits.TRISC7
#else
#define PWM4_TRIS TRISGbits.TRISG3
#endif

```

```

#if defined CC3_IO_V1
#define PWM3_TRIS TRISDbits.TRISD1
#elif defined CC8_IO_V1
#define PWM3_TRIS TRISCbits.TRISC6
#elif defined CC8_IO_V2
#define PWM3_TRIS TRISCbits.TRISC6
#else
#define PWM3_TRIS TRISGbits.TRISG0
#endif

```

```

#if defined CC1_IO_V1
#define PWM1_TRIS TRISBbits.TRISB3
#elif defined CC1_IO_V2
#define PWM1_TRIS TRISCbits.TRISC5
#elif defined CC8_IO_V1

```

```
#define PWM1_TRIS  TRISBbits.TRISB4
#elif defined CC8_IO_V2
#define PWM1_TRIS  TRISDbits.TRISD4
#else
#define          PWM1_TRIS  TRISCbits.TRISC2
#endif
```

```
#if defined CC8_IO_V1
#define PWM2_TRIS  TRISCbits.TRISC2
#elif defined CC8_IO_V2
#define PWM2_TRIS  TRISCbits.TRISC2
#else
#define PWM2_TRIS  TRISEbits.TRISE7
#endif
```

```
#if defined CC4_IO_V2
#define PWM5_TRIS  TRISBbits.TRISB5
#elif defined CC8_IO_V1
#define PWM5_TRIS  TRISBbits.TRISB5
#elif defined CC8_IO_V2
#define PWM5_TRIS  TRISBbits.TRISB5
#else
#define PWM5_TRIS  TRISGbits.TRISG4
#endif
```

```
#if defined CC4_IO_V2
#define PWM6_TRIS  TRISBbits.TRISB6
#else
#define PWM6_TRIS  TRISEbits.TRISE6
#endif
```

```
#if defined CC4_IO_V2
#define PWM7_TRIS  TRISBbits.TRISB7
#else
#define PWM7_TRIS  TRISEbits.TRISE5
#endif
```

```
#if defined CC4_IO_V2
#define PWM8_TRIS  TRISCbits.TRISC1
#else
#define PWM8_TRIS  TRISEbits.TRISE4
#endif
```

```
#if defined CC4_IO_V2
#define PWM9_TRIS  TRISCbits.TRISC6
#else
#define PWM9_TRIS  TRISEbits.TRISE3
#endif
```

```
#if defined CC4_IO_V2
#define PWM10_TRIS  TRISCbits.TRISC7
```

```
#else
#define PWM10_TRIS  TRISEbits.TRISE2
#endif

#endif
```